

TEKNIIKAN JA LIIKENTEEN TOIMIALA

Tietotekniikka

Tietoliikennetekniikka

INSINÖÖRITYÖ

TURVA- JA OPASTEVALOJEN RYHMÄVALVONTAKESKUS

**Työn tekijä: Timo-Mikael Sivula
Työn valvoja: Timo Karilinna
Työn ohjaaja: Pekka Sjöman**

Työ hyväksytty: __. __. 2006

**Timo Karilinna
lehtori**

ALKULAUSE

Tämä insinöörityö tehtiin projektiluontoisena SF-Design Oy:n nimissä sen asiakkaalle. Haluan kiittää projektissa apuna ja neuvojina olleita työn ohjaajaa diplomi-insinööri Pekka Sjömania ja diplomi-insinööri Nestori Fabritiusta. Tuon kiitokseni myös työn valvojalle diplomi-insinööri Timo Karilinnalle. Kiitän myös vaimoani ja perhettäni, jotka olivat projektissa henkisenä tukena ja jaksoivat aina kannustaa jaksamaan pitkienkin päivien jälkeen.

Forssassa 21.09.2006

Timo-Mikael Sivula

INSINÖÖRITYÖN TIIVISTELMÄ

Tekijä: Timo-Mikael Sivula	
Työn nimi: Turva- ja opastevalojen ryhmävalvontakeskus	
Päivämäärä: 21.09.2006	Sivumäärä: 34 s. + 3 liitettä
Koulutusohjelma: Tietotekniikka	Suuntautumisvaihtoehto: Tietoliikennetekniikka
<p>Työn valvoja: diplomi-insinööri Timo Karilinna</p> <p>Työn ohjaaja: diplomi-insinööri Pekka Sjöman</p>	
<p>Tämä insinöörityö on dokumentti projektista, joka on tehty SF-Design Oy:n nimissä asiakasprojektina. Työssä käydään läpi turva- ja opastevalojen ryhmävalvontayksikön elektroniikkasuunnittelu ja siihen tarvittavan mikrotietokoneen ohjaamiseen tarvittavan ohjelman suunnittelu ja kirjoittaminen. Työ jakautuu karkeasti jaoteltuna neljään osa-alueeseen. Ensiksi käydään läpi laitteiston elektroniikkasuunnittelu ja siihen liittyvää teoriaa. Seuraavaksi työssä käsitellään mikrotietokoneiden ohjelmistokehitystä ja käydään läpi työssä suunniteltu ja laadittu ohjelmisto STMicroelectronicsin ST62T10C-prosessorille. Lopuksi ennen suunnittelu- ja kehitysympäristöjen esittelyä käsitellään suunnittelun laitteiston testaus-, käyttö- ja asennustoimenpiteet.</p> <p>Aivan aluksi esitellään projektissa toteutettuun laitesuunnitteluun vaikuttaneet asiakkaan antama spesifikaatio ja 13.2.2003 voimaan astunut Euroopan parlamentin ja neuvoston asettama RoHS-direktiivi. Direktiivi vaikutti elektroniikkasuunnitteluun komponenttien valinnan osalta. Direktiivin mukaan sen voimaantumispäivän jälkeen ei saa käyttää komponentteja, jotka sisältävät ympäristölle ja ihmisille vaarallisia aineita.</p> <p>Elektroniikkasuunnitteluosiossa suunnittelu on jaettuna toiminnallisiin osiin. Osiossa käydään läpi virtamuuntajien kalibrointi sekä jännitevahvistimen suunnittelu ja simulointi. Kyseisessä osiossa käsitellään vielä prosessorin vaatimat erityiskytkennät, vikailmoitus-led-valojen ohjaukseen tarvittava kytkentä sekä ennen laitteiston vaatimien käyttöjännitteiden suunnittelua laitteistoon tarvittavien potentiovapaitten lähtöjen toteutus. Ohjelmistokehitysosiossa pohditaan mikrotietokoneen ohjelmoinnin perusteita, ennen kuin projektissa laadittua ohjelmaa käydään läpi. Projektissa laadittua koodia ei sen salaisuuden takia käydä yksityiskohtaisesti läpi tässä työssä.</p> <p>Projektissa käytettiin elektroniikkasuunnittelussa PADS-ohjelmistoja PowerLogic ja PowerPCB. Lisäksi käytettiin vahvistinkytkennän simulointiin OrCADin Capture CIS -ohjelmaa. Projektin työläimpään vaiheeseen, ohjelman suunnitteluun, kehitykseen ja kirjoittamiseen käytettiin Raisonancen STMicroelectronicsin mikrotietokoneille kehittämää ohjelmistokehitysympäristöä RIDE -ohjelmistoa. Elektroniikkasuunnittelussa käytetyt ohjelmistot ovat hyvin yleisesti teollisuuden käytössä.</p>	
Avainsanat: sulautettu järjestelmä, mikrotietokone, assembler, RoHS	

ABSTRACT

Name: Timo-Mikael Sivula	
Title: Group Supervision Unit of Guide and Security Lights	
Date: 21.09.2006	Number of pages: 34
Department: Information Technology Study Programme: Telecommunications	
Instructor: Timo Karilinna, M.Sc.	
Supervisor: Pekka Sjöman, M.Sc.	
<p>This written report is a document of project which was carried out in a customer project of SF-Design Oy. The purpose of this study was to examine the planning of the electronics of a group supervision unit of guide and security lights and to plan and write codes for the microcomputer program. The study was carried out by discussing the theory and planning of electronics equipment. One aim was to describe the software development of microcomputers and introduce the planned and written program for the equipment planned within this study. Another objective was to explain the planning and development environments as well as the testing, use and installation of the planned equipment.</p> <p>One important factor affecting the planning of the electronics equipment was the RoHS directive of the European parliament and council. According to the directive it is forbidden - after the directive has become effective - to use components which include dangerous materials for humans and environments. Another factor affecting the planning was the equipment specification from the customer.</p> <p>The chapter on electronics planning discusses the different functional parts of electronics planning. These include the calibration of current transformers, planning and simulation of voltage amplifier, special connections of processors, connection for controlling the led lights, voltage-free outputs of equipment and planning of voltage use. The next chapter on program development and the theory of microcomputers also explains that program that was written in this project. The code is, however, secret and is therefore not explained in detail.</p> <p>When planning the electronics, PowerLogic and PowerPCB software which includes PADS software were used. For the simulation of the amplifier connection Capture CIS software from OrCAD was used. The software employed in the electronics planning is widely used within the industry. In the software development stage, RIDE software manufactured by Raisonance. This software has been developed for the STMicroelectronics microcomputers.</p>	
Keywords: Embedded System, Microcomputer, assembler, RoHS,	

SISÄLLYS

ALKULAUSE

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	1
2	VALVONTAKESKUKSEN SPESIFIKAATIO	2
3	SUUNNITTELUPROSESSI	5
3.1	Virtamuuntajan kalibrointi	5
3.2	Vahvistuskytkennän simulointi	6
3.3	Prossessorin kytkennät	9
3.4	LED-ohjauskytkentä	11
3.5	Potentiovapaat lähdöt	12
3.6	Käyttöjännitteet	13
3.7	Yhteenveto	14
4	OHJELMISTOKEHITYS	15
4.1	Yleistä mikrotietokoneen toiminnasta	15
4.2	Yleistä ohjelmoinnista	19
4.3	Vuokaavio	21
4.4	Koodin kirjoittaminen	22
5	LAITTEISTON TESTAUS	25
6	LAITTEISTON ASENNUS JA KÄYTTÖ	27
6.1	Asennus	27
6.2	Käyttö	28
7	SUUNNITTELUYMPÄRISTÖT	29
7.1	RIDE	30
7.2	Winee	32
8	YHTEENVETO	33
	VIITELUETTELO	34

LIITTEET

Liite 1. EXICMM -piirikortin piirikaaviot

Liite 2. EXICMD -piirikortin piirikaavio

Liite 3. Assembler-koodin vuokaavio

1 JOHDANTO

Tämän insinööriyön lähtökohtana oli suunnitella ryhmävalvontakeskus, jolla pystytään keskitetysti testaamaan turva- ja opastevalojen toimivuus.

Työtä lähdettiin suunnittelemaan asiakkaan toimittaman spesifikaation mukaisesti STMicroelectronicsin ST6-prosessoriperheeseen kuuluvan ST62T10C6-prosessorin ympärille. Laitteiston täytyi täyttää Euroopan parlamentin ja neuvoston RoHS (*the Restriction of use of certain Hazardous Substances*) -direktiivi ja olla muutenkin valmistuskustannuksiltaan mahdollisimman edullinen.

Tässä työssä käydään läpi laitteiston suunnitteluperusteet sekä suunnittelu- ja toteutusvaiheet aina lopputestaukseen asti. Työssä esitellään vain ohjelmistosuunnitteluympäristö, koska muut käytetyt suunnitteluympäristöt ovat yleisesti teollisuuden käytössä. Projekti sisälsi sekä elektroniikkasuunnittelua että assembler-ohjelmointia. Mikrotietokoneen ohjelmointi oli projektissa suurin kokonaisuus, joten siihen paneudutaan hiukan tarkemmin luvussa 4.

Elektroniikkasuunnittelu on jaettu tässä työssä toiminnallisten lohkojen mukaisesti. Eri lohkojen suunnittelua läpikäydessä kerrotaan myös komponenttien valintaan vaikuttaneet asiat. Elektroniikkasuunnittelun perusteita ei käydä työssä läpi syvällisesti. Operaatiovahvistinten teoriasta on selvitetty niiden peruskytkentöjen perusteet, mutta elektroniikkasuunnitteluun vaikuttaneet asiat, laitteiston spesifikaatio ja RoHS-direktiivi esitellään tarkemmin.

Mikrotietokoneen ohjelmoinnin perusteita käydään läpi tarkemmin ja syvällisemmin paneutuen mikrotietokoneen rakenteeseen ja sisäiseen toimintaan. Yleistä mikrotietokoneen ohjelmistokehityksen kulkua käsitellään samalla kun projektin aikana toteutettua ohjelmistokehitystäkin. Työn lopuksi esitellään tarkemmin ohjelmistokehitykseen käytetyt suunnittelu- ja kehitysympäristöt. Elektroniikkasuunnittelun työkaluja ei esitellä tarkemmin niiden yleisen käytön takia.

2 VALVONTAKESKUKSEN SPESIFIKAATIO

Asiakas toimitti spesifikaation, jonka mukaisesti tehtäväksi annettua suunnittelua lähdettiin viemään eteenpäin. Seuraavaksi on esitelty kyseinen spesifikaatio.

Yleistä

Suunniteltavan piirikortin on tarkoitus toimia EXI24DC-piirikortin ulostulojen virran mittauksessa. Ulostuloista (8 kpl) mitataan virtaa jatkuvasti ja tehdään hälytys, jos jokin ulostulon virta muuttuu huomattavasti. Suunniteltavassa laitteessa on asetusnappi, jota painamalla suoritetaan alkumittaus, johon jatkuva-aikaista mittausta verrataan.

Spesifikaatiot

- ?? Valvontakeskuksessa on 16 ryhmää, jotka mittaavat valaistusryhmien virtaa.
- ?? Ryhmille annetaan asetteluarvo painamalla testausnappia 10 sekunnin ajan. Valaistusryhmän virtaa verrataan kyseiseen asetteluarvoon. Kun valaistusryhmän ottama virta muuttuu eli jokin valaisin ei toimi, antaa tämä ryhmä hälytyksen. Ryhmällä on merkki-led keskuksen kannessa.
- ?? Keskuksessa on painike, josta voidaan haluttaessa tehdä testaus sekä antaa asetteluarvot prosessorille. Testattaessa turvalokeskus käynnistyy ja mittaus tehdään ns. verkkohäiriötilassa. Samalla tulee tehtyä turvalokeskuksen testaus.
- ?? Keskus tekee automaattisesti kuukausittain testin, jossa turvalokeskus käynnistyy ja keskus mittaa ryhmien virta-arvot. Samalla tulee tehtyä standardeissa vaadittu kuukausitestausta, keskus pysyy päällä n. 10 minuuttia, jonka jälkeen se palautuu normaalitilaan.
- ?? Vikatilanteessa annetaan ryhmien yhteishälytys potentiovapaalla lähdöllä, esim. releen välityksellä.
- ?? Jokaisesta ryhmästä on oma led-näyttö kannessa.
- ?? Ryhmien kaapelointi viedään suoraan turvalokeskukseen normaalisti.
- ?? Valvontakeskukseen tulee virtamuuntajat. Valvontakeskus liitetään lattaakaapelilla valvontakorttiin.

- ?? Käyttöjännite on 24 V DC $I_{\max}=500$ mA.
- ?? Sisäänmenoina 8 kpl virtamittaussilmukoita, joiden läpi EXI24DC:n lähdöt johdetaan.
- ?? Ulostuloina 8 kpl indikaatioledejä, joista näkee missä ryhmässä on vikatilanne.
- ?? Ulostulo 1 kpl potentiovapaa optoeroitettu lähtö, josta lähtee hälytys keskukselle.
- ?? Ulostulo 1 kpl potentiovapaa optoerotettu lähtö, jolla annetaan EXI24DC piirikortille ilmoitus että sen pitää käynnistyä.
- ?? Piirilevyn ulkomitat sovitetaan haluttuihin mittoihin.
- ?? Laitteesta pyritään tekemään mahdollisimman edullinen niin komponenttien kuin valmistuskustannustenkin osalta.
- ?? Laite toteutetaan läpiladottavilla komponenteilla.

Edellä oleva spesifikaatio ja RoHS-direktiivi antoivat hyvän pohjan suunnittelun aloitukselle. Edellä lueteltu spesifikaatio tarkentui joiltakin osilta työn edetessä. Spesifikaatiossa mainittu EXI24DC-kortti on asiakkaan turva- ja opastevalojen ohjausyksikön piirikortti. Kahden indikaatioledin lisävaatimus spesifikaatioon aiheutti sen, että prosessori ei soveltunut täysin edellä olevaan spesifikaatioon. Tämän takia jouduttiin poikkeamaan spesifikaatiosta esimerkiksi lisäämällä yksi kytkin lisää. Suunnittelun etenemisestä on kerrottu tarkemmin luvussa 3.

Yleisellä tasolla laitteiston toiminta on seuraavanlainen.

- ?? Muunnetaan virta jännitteeksi.
- ?? Vahvistetaan saatua jännitettä.
- ?? Tehdään jännitteelle A/D-muunnos.
- ?? Kalibroitaessa tallennetaan muunnoksen tulos.
- ?? Testattaessa verrataan muunnoksen tulosta muistissa olevaan tietoon.
- ?? Testattaessa annetaan hälytys, jos on vikaa kanavissa.
- ?? Muu aika suoritetaan aikalaskentaa, joka perusteella suoritetaan automaattinen testaus.

RoHS-direktiivi

Koska Euroopan parlamentti ja neuvosto on asettanut direktiivin 2002/95/EY (myöh. RoHS-direktiivi) ja laitteiston markkinoille tuloajankohta on RoHS-

direktiivissä mainitun 1.7.2006 päiväyksen jälkeen, täytyi laitteiston suunnittelussa ottaa huomioon kyseinen direktiivi. Direktiivi tuli voimaan 13.2.2003 ja sen soveltaminen on aloitettu 1.7.2006. Seuraavassa lainaus RoHS-direktiivin 1. artiklasta, josta käy ilmi direktiivin asettamisen tarkoitus.

Tällä direktiivillä on tarkoitus lähentää jäsenvaltioiden lakeja, jotka koskevat vaarallisten aineiden käytön rajoittamista sähkö- ja elektroniikkalaitteissa, sekä edistää ihmisten terveyden suojelua ja sähkö- ja elektroniikkalaiteromun hyödyntämistä ja loppukäsittelyä ympäristöä säästävällä tavalla [1. s. 20].

Direktiiviä tulee toisen artiklan mukaan pääsääntöisesti soveltaa kaikkiin sellaisiin sähkö- ja elektroniikkalaitteisiin ja niiden varaosiin, jotka ovat tulleet markkinoille 1.7.2006 jälkeen rajoittamatta millään tavalla turvallisuus- ja terveysvaatimuksia koskevaa yhteisön lainsäädäntöä sekä jätehuoltoa koskevaa yhteisön erityislainsäädäntöä.

Direktiivin neljännen artiklan mukaan EU:n jäsenvaltioiden on varmistettava, että mainitun päivämäärän jälkeen markkinoille tulevat sähkö- ja elektroniikkalaitteet eivät sisällä lyijyä, elohopeaa, kadmiumia, kuudenarvoista kromia, polybromibifenyylä (PBB) ja/tai polybromidifenyylieetteriä (PBDE). Edellä mainittua neljännen artiklan asiaa ei sovelleta muutamiin käyttötarkoituksiin, jotka ovat lueteltuna direktiivin liitteessä. Seuraavaksi kuitenkin niistä muutama esimerkki:

?? elohopea pienoislampeluissa silloin, kun elohopean määrä lamppua kohden on enintään 5 mg

?? elohopea erityiskäyttöön tarkoitetuissa loisteputkissa

?? lyijy katodisädeputkien lasissa, elektroniikkakomponenteissa ja loisteputkissa

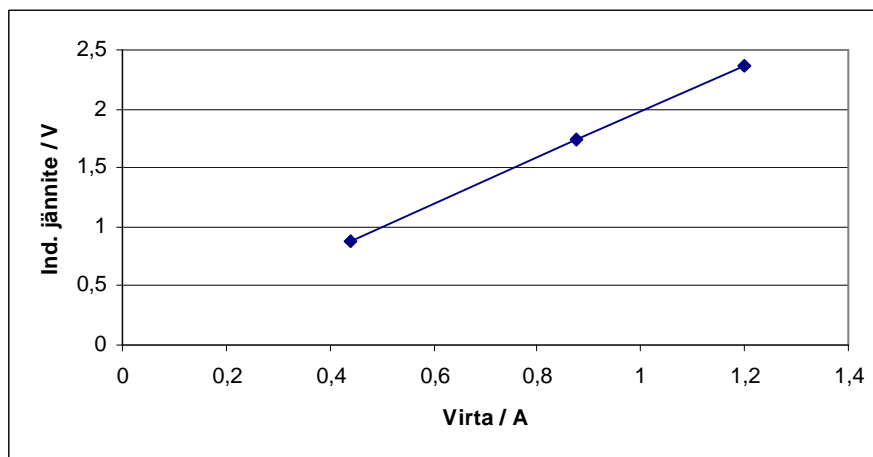
Komponenttien toimittajat tiedottavat tuotteidensa RoHS-sopivuudesta erittäin hyvin, joten komponenttien sisältämiä aineita ei tämän takia tarvitse itse selvittää.

3 SUUNNITTELUPROSESSI

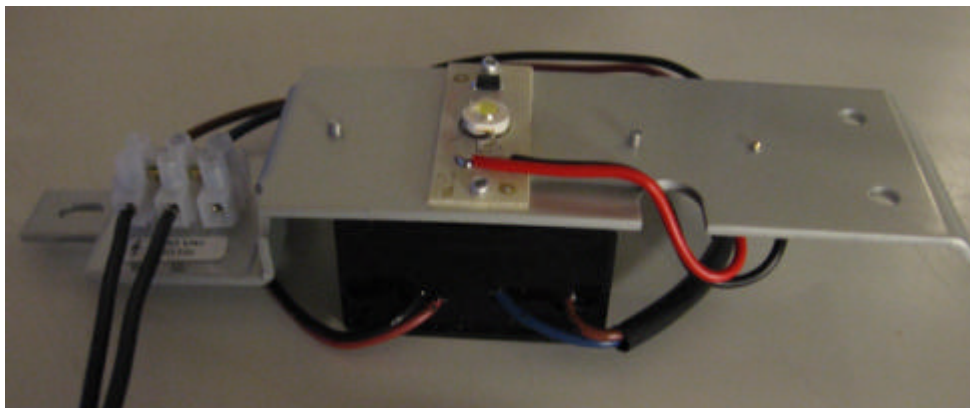
Tässä luvussa käydään läpi suunnitteluprosessi elektroniikkasuunnittelun osalta sekä komponenttien valintaperusteet. Elektroniikkasuunnittelu on jaettu kuuteen toiminnalliseen osioon. Elektroniikkaosioon liittyvä teoria on sisällytetty kyseisiin osioihin. Ohjelmistosuunnittelu on luvussa 4.

3.1 Virtamuuntajan kalibrointi

Suunnittelu aloitettiin mittaamalla spesifikaatiossa mainittujen virtamuuntajien indusoiman jännitteen suuruutta virran funktiona, jonka avulla pystyttiin määrittelemään yksikön dynamiikka ja mittaustarkkuus. Mittaukset suoritettiin Electrohmsin RD189-virtamuuntajalla, joka oli asiakkaan valinta virtamuuntajaksi. Näiden mittausten perusteella piirrettiin kuvaaja, jonka avulla oli helppo määritellä tarvittava vahvistus indusoituneelle jännitteelle. Kuvassa 1 on esitetty kyseisen mittauksen tulokset. Mittauksessa selvitettiin myös se, kuinka paljon yhden opaste- ja turvavalon kuluttama virta indusoi jännitettä. Kuvassa 2 olevan edellä mainitun lampun kuluttama virta indusoi noin 30 mV jännitteen.



Kuva 1. Virtamuuntajan indusoima jännite virran funktiona

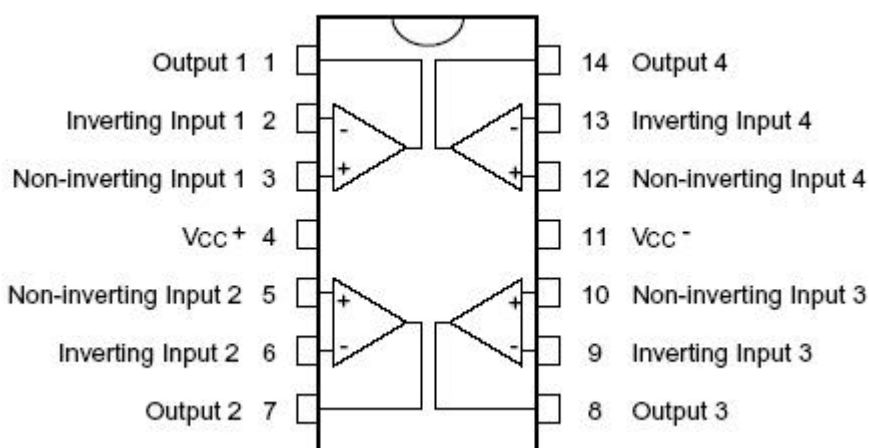


Kuva 2. Opaste- ja turvavalo

Kun asiakkaan ilmoituksen mukaan yhden mitattavan kanavan päässä on maksimissaan 20 kuvassa 2 olevaa lamppua, voitiin kuvassa 1 olevan käyrän avulla määritellä se, kuinka paljon virtamuuntajan indusoimaa jännitettä voidaan vahvistaa. Kuvassa 2 olevan lampun valmistaja ilmoittaa lampun kuluttavan 20 mA virtaa, joten yhden kanavan maksimi virraksi tulee 400 mA.

3.2 Vahvistuskytkennän simulointi

Kun saatiin tietoon kanavan maksimi virta, voitiin kuvasta 1 katsoa kuinka paljon maksimissaan saadaan indusoitua jännitettä virtamuuntajalta. Näillä tiedoilla pystyttiin aloittamaan jännitteelle tarvittavan vahvistuksen suunnittelu. Vahvistuskytkentää suunniteltiin Texas Instrumentsin valmistaman JFET (junction gate field-effect transistor) -operaatiovahvistin TL074CN-piirin ympärille. Piirin toiminnallinen kuva on kuvassa 3. DIL-16 kotelo valittiin, koska siinä on 4 kappaletta operaatiovahvistinta yhdessä piirissä, sekä valintaan vaikutti myös piirin alhainen kohina ja tarpeeksi suuri stabiili vahvistus. [3.]



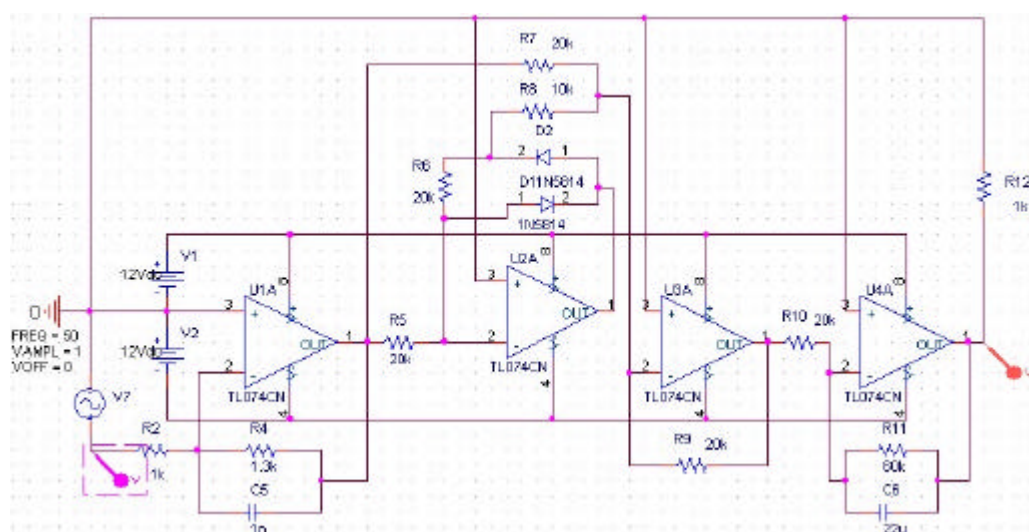
Kuva 3. TL074CN-piirin toiminnallinen kuva

Operaatiovahvistin on useammasta transistorista rakennettu vahvistin, jonka vahvistus on tavallisesti hyvin suuri. Vahvistimen tuloasteena on käytetty differentiaaliastetta. Operaatiovahvistimella on kaksi tuloastaa: kääntävä eli invertoiva ja ei-kääntävä eli ei-invertoiva tulonasta. Operaatiovahvistin vahvistaa tulojännitteidensä erotusta. Negatiivista takaisinkytkentää käytettäessä operaatiovahvistin pyrkii pitämään lähtönsä lineaarisella toiminta-alueella. Positiivisessa takaisinkytkennässä operaatiovahvistimen lähtö pystyy vaikuttamaan ei-invertoivan tulonastan potentiaaliin. Tällöin lähtö on joko negatiivisessa tai positiivisessa laidassa riippuen siitä, kumpi tulojännitteistä on suurempi. Kun positiivisen tulonastan jännite on suurempi kuin negatiivisen tulonastan jännite, operaatiovahvistimen lähtö on positiivisessa laidassa. Lähtö on negatiivisessa laidassa, kun positiivisen tulonastan jännite on pienempi kuin negatiivisen tulonastan jännite. Positiivisessa takaisinkytkennässä operaatiovahvistin toimii digitaalipiirin tavoin. Kun operaatiovahvistimen negatiivinen tulonasta kytketään suoraan maatasoon, saadaan varmistettua se, että lähtöjännite on aina joko negatiivisessa tai positiivisessa laidassa. Koska positiivisessa takaisinkytkennässä lähtöjännite ei ole lineaarisella toiminta-alueella, tuloastojen jännitteet eivät koskaan ole yhtä suuria. [9, s.93 - 116, 471 - 576 ja 10, s.104 - 109]

Vahvistinkytkentää simuloitiin OrCADin Capture CIS -ohjelman avulla. Kyseiseen vahvistuskytkentään sisällytettiin myös tasasuuntausosio, sillä prosessorille voitiin syöttää DC-jännitettä ja virtamuuntajalta saatiin 50 Hz AC-jännitettä.

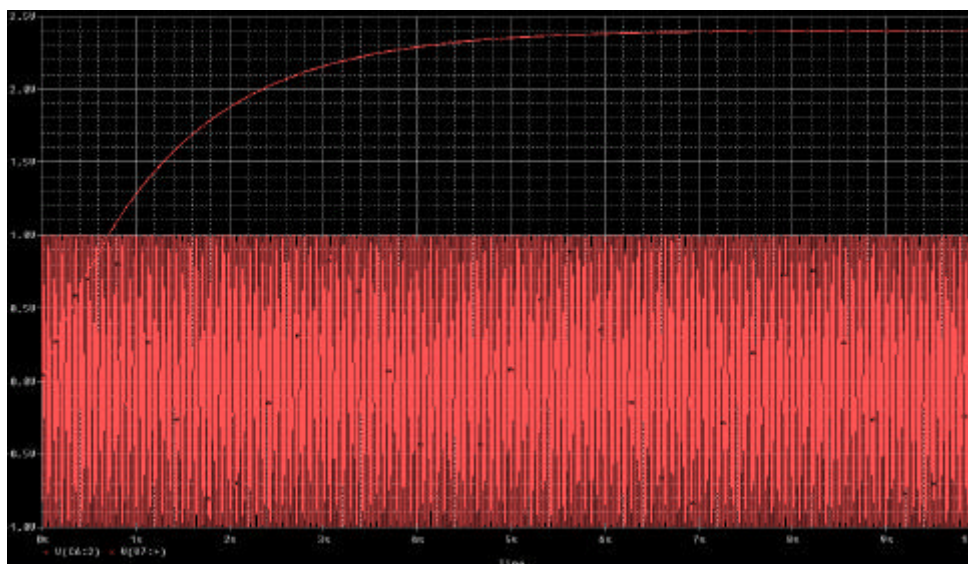
Vahvistusasteen suunnittelun lähtökohdaksi otettiin tilanne, jossa yhden EXI24DC-piirin kanavassa on 50 lamppua. Tämä tarkoitti sitä että maksimi virta on 1 A ja silloin saadaan virtamuuntajalta 2 voltin AC-jännite. Näin ollen jännitettä voidaan vahvistaa parhaan dynamiikan saamiseksi kaksi ja puolikertaiseksi, prosessorin sisäänmenojännitteen ollessa 5 voltia. Vahvistamisen lopuksi jännite pitää ohjata Chebyshev-topologiolla toteutetun aktiivisen integroivan yksiasteisen suodattimen läpi, jotta saadaan suodatettua loput mahdolliset häiriöt pois.

Kytkenä on esitetty kuvassa 4. Kytkenässä olevat operaatiovahvistinkomponentit esittävät yhdessä valittua TL074CN-piiriä. Kytkenässä on neljä vahvistusastetta, joissa on kaikissa negatiivinen takaisinkytkentä. Ensimmäisessä asteessa olevan vastuksen ja kondensaattorin muodostaman ensimmäisen asteen suodattimen avulla toteutettu takaisinkytkentä suodattaa osan häiriöistä, joita virtamuuntajalta tulee. Toiseen asteeseen on sijoitettu tasasuuntausdiodeit, joiden avulla saatua DC-signaalia integroidaan neljännessä asteessa olevan $22\mu\text{F}$ kondensaattorin ja vastuksen rinnankytkennän avulla aikavakion ollessa 1.32. Tämä on otettava huomioon tehtäessä kanaville mittauksia. Neljännessä asteen ulostulosta saadaan integroitu DC-jännite, jota on vahvistettu noin kaksi ja puolikertaiseksi. Näin saatu signaali voitiin syöttää suoraan prosessorin analogiatuloihin, porttiin B. Tällainen vahvistinkytkentä laitettiin jokaisen kahdeksan virtamuuntajan ja prosessorin väliin.



Kuva 4. Vahvistinkytkennän simulointikytkenä

Vahvistuskytkennän alkuun tulovastuksen rinnalle lisättiin vielä 4.7 voltin zenerdiodi ylijännitesuojaksi suojaamaan laitteistoa mahdolliselta jännitepiikiltä. Kuvassa 5 on simulointitulos kuvan 4 simulointikytkennästä. Simulointiaika on 10 sekuntia ja jänniteskaala on -1 voltista 2.5 volttiin. Simuloinnissa on sisään syötetty 1 voltin 50 Hz:n sinisignaalia.



Kuva 5. Simulointitulos

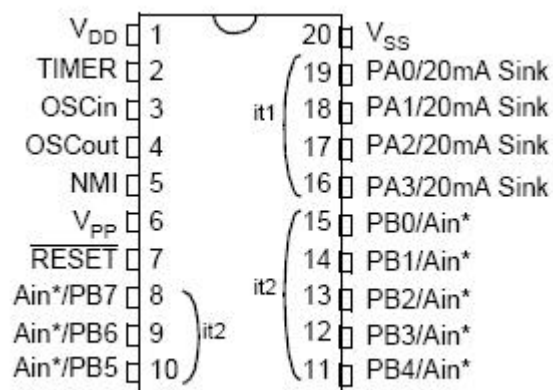
Operaatiovahvistin kytkentää simuloitaessa yksinkertaisin tapa muuttaa kytkennän vahvistusta on muuttaa viimeisen operaatiovahvistinasteen takaisinkytkennässä olevaa vastusta. Kytkentää simulointiin usealla eri sisääntulojännitteellä. Kuvassa 4 olevilla komponenttiarvoilla neljännen asteen integrointiajaksi saatiin 6 sekuntia. Mitä suurempi vahvistus halutaan, sitä pidempi on integrointiaika. Tämä tarkoittaa sitä, että kun kanavia kalibroidaan tai testataan, on odotettava että kyseinen aika kuluu ennen kuin A/D-muunnosta voidaan tehdä. Simuloitaessa selvitettiin kuormavastuksen vaikutusta vahvistukseen. Kyseisellä vastuksella ei ollut mitään vaikutusta vahvistukseen, joten sillä minkälaiseen kytkentään vahvistettu jännite syötetään, ei ole vaikutusta vahvistukseen.

3.3 Prosessorin kytkennät

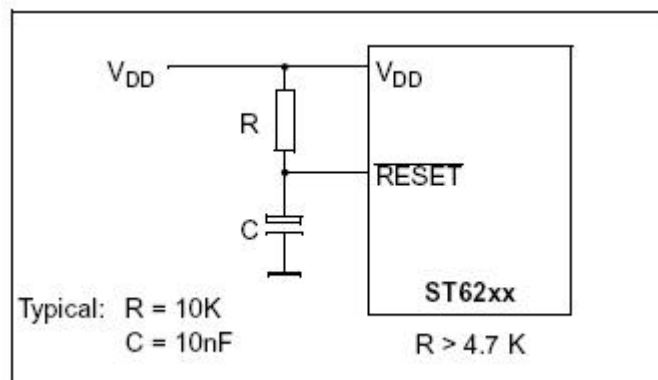
Kun vahvistus- ja tasasuuntauskytkentä saatiin halutun kaltaiseksi, seuraavaksi suunniteltiin prosessorin tarvitsemia ulkoisia kytkentöjä (kuva 6). Prosessorin vaatimat kytkennät toteutettiin sen datalehden mukaisesti.

Erityisesti prosessorin RESET-pinni vaati omanlaisensa kytkennän, joka on esitetty kuvassa 7. Tällöin voidaan käyttää tarvittaessa sisäisiä LVD (*Low Voltage Detector*) ja vahtikoira reset -toimintaa. Myös prosessorin ulkoinen kellokytkentä on datalehdessä määritelty. Prosessorille valittiin kelloksi kide, jonka taajuus oli datalehden mukaisesti 8.00 MHz ja kyseinen kytkentä toteutettiin myös datalehden mukaisesti. Kuvassa 8 on esitetty kiteellä toteutettu kellokytkentä. Kellotaajuuden stabiilisuuden takaamiseksi kelloksi valittiin kide. Valitun kiteen taajuuden stabiilisuudeksi valmistaja ilmoittaa maksimissaan ± 100 ppm [4, s. 3]. Laitteiston spesifikaatiossa on mainittu, että laitteen on tehtävä kerran kuukaudessa automaattinen testi. Kiteen stabiilisuus on testin ajoittamisen kannalta hyvin tärkeä. Seuraavassa on esitelty laskelma, kuinka paljon taajuuden vaihtelu voi vaikuttaa testin ajoitukseen.

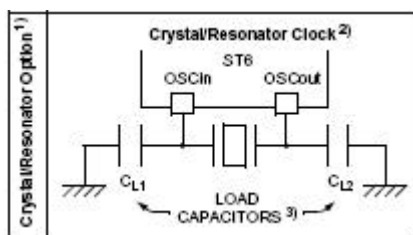
- ?? Laskennan aikana yhden käskyn suorittamiseen kuluu $3,333$ kellojaksoa.
- ?? Ohjelmallisesti toteutettu kuukauden kestävä laskenta sisältää $6,88060737 \cdot 10^{12}$ suoritettavaa käskyä.
- ?? Käskyjen suorittamiseen kuluu $2,2935358 \cdot 10^{13}$ kellojaksoa.
- ?? Jos kiteen taajuuden virhe on maksimi jonka valmistaja ilmoittaa mahdolliseksi, eroaa sillä arvolla laskettu suoritusaika teoreettisella arvolla lasketusta suoritusaikasta $\pm 35,84$ sekuntia.



Kuva 6. Mikrotietokoneen ulkoiset kytkennät [2, s. 7]



Kuva 7. Prosessorin RESET -pinnin ulkoinen kytkentä [2, s. 26]



Kuva 8. Prosessorin kellokytkentä [2, s. 20]

Prosessorin A-portin pinnit valittiin spesifikaatiossa mainittujen merkkiled-valojen ja yhteishälytyksen ohjaamiseen. Datalehden mukaan saatiin prosessorin TIMER-pinni ohjaamaan turvalokeskuksen käynnistymistä. TIMER-pinni ei tarvinnut mitään erityistä kytkentää, vaan sen toiminta hoidettiin ohjelmallisesti. Prosessorin B-portin kaikkiin 8 pinniin kytkettiin vahvistetut jännitesignaalit.

3.4 LED-ohjauskytkentä

Koska prosessorin portteja voidaan ohjata ohjelmallisesti, saatiin A-portin näljällä pinnillä ohjattua kahdeksaa led-valoa. Led-valojen ohjaukseen jouduttiin lisäksi valitsemaan IC-piiri, jolla saadaan muutettua kolme digitaalista signaalia, joita ohjataan ohjelmallisesti, kahdeksaksi digitaaliseksi signaaliksi. Piiriksi valittiin Philips Semiconductorsin valmistama 3:sta 8:aan linjadekooderi SN74HC138N.

Taulukossa 1 on esitetty, kuinka valittua piiriä voidaan ohjata neljällä digitaalisella signaalilla. Prosessorin A-portin lähdöt kytkettiin taulukossa 2 näkyviin dekooderin pinneihin E3, A2, A1 ja A0. Näiden neljän tulon muodostamaa bittikuviota muuttamalla saatiin piirin lähdöt vaihtumaan taulukon 2 osoittamalla tavalla.

Taulukosta 1 nähdään, että led-valot voidaan kytkeä suoraan vastuksen kanssa sarjankytkettyinä piirin lähtöihin. Led-valojen toiselle puolelle tuodaan korkea jännitetaso eli 5 voltin jännite. Led syttyy silloin kun SN74HC138N-piirin lähtö asetetaan alas. Kytkentään laitettiin myös yksi led ilmoittamaan siitä että kaikki on kunnossa. Tämä led lisättiin sarjavastuksen kanssa SN74HC138N-piirin E3 tulopinnin ja 5 voltin väliin, joten kyseinen led palaa silloin kun E3 pinnille tuleva signaali on matala jännitetaso eli 0 voltia.

Taulukko 1. SN74HC138N-piirin toimintataulukko [5, s. 5]

Control			Input			Output							
$\bar{E}1$	$\bar{E}2$	E3	A2	A1	A0	$\bar{Y}7$	$\bar{Y}6$	$\bar{Y}5$	$\bar{Y}4$	$\bar{Y}3$	$\bar{Y}2$	$\bar{Y}1$	$\bar{Y}0$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X											
X	X	L											
L	L	H	L	L	L	H	H	H	H	H	H	H	L
			L	L	H	H	H	H	H	H	H	L	H
			L	H	L	H	H	H	H	H	L	H	H
			L	H	H	H	H	H	H	L	H	H	H
			H	L	L	H	H	H	L	H	H	H	H
			H	L	H	H	H	L	H	H	H	H	H
			H	H	L	H	L	H	H	H	H	H	H
			H	H	H	L	H	H	H	H	H	H	H

[1] H = HIGH voltage level;
L = LOW voltage level;
X = don't care.

3.5 Potentiovapaat lähdöt

Laitteiston spesifikaatiossa on kaksi potentiovapaata lähtöä, toinen yhteishälytystä varten ja toinen turvalokeskuksen ohjausta varten. Lähtöihin valittiin Good Sky Electric Co. Ltd:n valmistamat 3 voltin releet.

Yhteishälytyksen lähtöä päätettiin ohjata SN74HC138N-piirin ohjaussignaaliella eli signaalilla, joka tulee pinnille E3. Tämä signaali asetetaan aina ylös, kun jossakin testattavassa kanavassa on vikatilanne. Koska tämä signaali tulee prosessorilta ja prosessori antaa 20 mA virtaa, joka ei riitä releen ohjaamiseen, tarvitaan väliin transistori ohjaamaan relettä. Releen ohjaamiseen tarvittava nimellinen virta on 66.7 mA [6, s. 2]. Transistoriksi valittiin Philips Semiconductorsin valmistama BSN304 N-kanavainen D-MOS (*Double diffused Metal Oxide Semiconductor*)

transistori. Transistorin valintaan vaikutti suuresti RoHS-direktiivi. Koska releen ohjaamiseen tarvittiin transistori, täytyi releet valita sen mukaisesti. Valittu transistorin drain-source jännitehäviö on 2 voltia, joten releelle saadaan 3 voltia käytettäessä 5 voltin syöttöjännitettä. [7, s. 4]

Turvavalokeskuksen ohjaamiseen tarvittavaa potentiovapaata lähtöä päätettiin ohjata prosessorin TIMER-pinnin avulla. Samasta syystä kuin yhteishälytyksen releen ohjauksessa, tarvittiin myös tälle releelle transistori ohjaamaan sitä. Prosessorin TIMER-pinniltä saadaan ohjelmallisesti tuotettua ylös asetettua signaalia.

3.6 Käyttöjännitteet

Viimeinen elektroniikkasuunnittelun vaihe oli laitteiston käyttöjännitesyötön suunnittelu. Tämän suunnittelun lähtökohtana oli 24 voltin sisääntulo ja operaatiovahvistimen käyttöjännite ± 18 voltti ja prosessorin käyttöjännite 3.6 voltista - 6 volttiin. [2, s. 64 ja 3, s. 2]

Prossessorin käyttöjännitteeksi valittiin 5V, jonka muodostaminen 24 voltin sisääntulojännitteestä hoituu yhdellä regulaattorilla. 5 voltin regulaattoriksi valittiin STMicroelectronicsin valmistama L7805CV-regulaattori. Regulaattorin valintaan vaikutti RoHS-direktiivi, regulaattorin ulostulovirran suuruus ja komponenttien oli oltava läpiladottavia. Prosessori antaa ulostulojännitteenä lähes käyttöjännitteen kanssa yhtä suuren jännitteen. Operaatiovahvistimet tarvitsevat sekä positiivisen että negatiivisen käyttöjännitteen. Operaatiovahvistimien tarvitsemien jännitteiden muodostamiseen valittiin RoHS-direktiivin vaikuttamana 8 voltin TCS:n valmistama TS7808CI-regulaattori ja National Semiconductorin valmistama LMC7660-jänniteinvertioija piiri. Jänniteinvertioijan valintaan vaikutti myös ulkoisten komponenttien vähäinen määrä. Regulaattori ja jänniteinvertioija vaikuttivat suuresti keskenään toistensa valintaan.

Jänniteinvertioijia jouduttiin kytkemään neljä kappaletta rinnakkain, jotta saatiin operaatiovahvistimille riittävästi tehoa [8, s. 2]. Yhden operaatiovahvistimen kuluttama teho on 680 mW [3, s. 2].

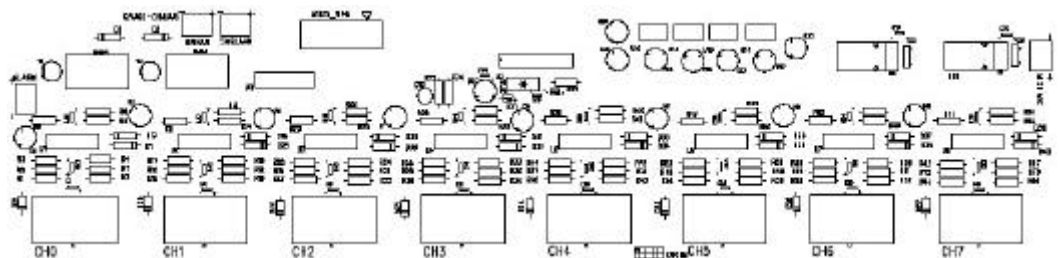
Jännitesyöttösuunnittelussa otettiin myös huomioon se että jokaiselle laitteistossa olevalle IC (*Integrated Circuit*) -piirille tulee sen vaatimat käyttöjännitteen suodatuskondensaattorit.

3.7 Yhteenveto

Laitteiston spesifikaation mukaisesti laitteistoon tuli kaksi piirikorttia, joista toinen on varsinaisesti itse laitteisto (EXICMM) ja toinen on pienempi ohjaus- ja valvontakortti (EXICMD). Ohjaus- ja valvontakortille sijoitettiin LED-valot kuormavastuksineen ja testaus- sekä kalibrointinapit indikaatioledeineen.

Kun jokainen erillinen osio oli suunniteltu, piirrettiin piirikaviot PADS-ohjelmistoihin kuuluvalla PowerLogic-ohjelmistolla. Piirikorttien piirikaaviot ovat liitteissä 1 ja 2. EXICMM-piirikaavion piirtämisessä käytettiin toiminnallisen ja jännitesyöttöosan erottamista erillisiin piirikaavioihin. Tämä selkeytti piirikaavioita suuresti.

Piirikaavioiden piirtämisen jälkeen piirrettiin piirikortit PADS-ohjelmistoihin kuuluvalla PowerPCB-ohjelmistolla. Piirikaavion ja piirikortin suunnitteluohjelmistojen välillä voidaan siirtää tehtyjä muutoksia kumpaankin suuntaan. Piirikaavion sisältämät tiedot siirrettiin piirikortin piirto-ohjelmaan, jossa tehtiin komponenttisijoittelu ja reititys. Reititys voitiin tehdä joko manuaalisesti tai automaattisella reititystyökalulla. Komponentit sijoitettiin toiminnallisiin lohkoihin selkeästi eroteltuna toisistaan. Kuvassa 9 on kuva piirikortin silkkipainosta eli kuva on piirikortin pinnalle painettava kuva. Piirikorteista tehtiin kaksikerroslevyjä, koska näin saatiin piirilevyn valmistus kustannuksen mahdollisimman pieneksi. Reitityksessä otettiin huomioon virtojen suuruudet signaalivetojen leveydessä. Reitityksessä pyrittiin saamaan piirikortin pohjapuoli mahdollisimman yhtenäiseksi toimivan maatason aikaansaamiseksi, joten reititys tehtiin suurimmaksi osaksi piirikortin yläpuolella.



Kuva 9. EXICMM-piirikortin silkkipainokuva

Spesifikaation mukaisesti komponenttien valinnassa onnistuttiin hyvin. Löydettiin halvat komponentit, jotka täyttivät myös RoHS-direktiivin.

4 OHJELMISTOKEHITYS

Tässä osiossa käydään läpi mikrotietokoneen ohjelmointiin liittyvää teoriaa, ennen kuin projektissa toteutettua ohjelmistokehitystä käydään läpi. Ensimmäisessä osiossa syvennytään mikrotietokoneen toimintaan, jonka tietämys on edellytys sen ohjelmoinnille. Toisessa osiossa selvitetään mikrotietokoneen ohjelmointiin liittyviä asioita yleisellä tasolla erittelemättä eri prosessorityyppejä. Kahdessa viimeisessä osiossa käydään toteutettu ohjelmistokehitys läpi sen salaisuuden sallimissa rajoissa.

4.1 Yleistä mikrotietokoneen toiminnasta

Mikrotietokoneet jaetaan yleensä kahteen ryhmään. Yleiskäyttöiset mikrotietokoneet ovat tarkoitettu henkilökohtaiseen tietojenkäsittelyyn ja sulautetut järjestelmät ovat tarkoitettu johonkin erityiseen toimintaan räätälöityihin mikrotietokonelaitteisiin. Yleiskäyttöiset mikrotietokoneet tunnetaan paremmin nimellä PC (*Personal Computer*). Elektroniikkajärjestelmää, jossa on mukana mikrotietokone, kutsutaan sulautetuksi järjestelmäksi. Tässä osiossa tarkastellaan mikrotietokoneen toimintaa sulautetun järjestelmän kannalta.

Sulautettuja järjestelmiä käytetään johonkin tiettyyn erityistarkoitukseen. Näiden laitteiden muisti jakautuu fyysisesti kahteen osaan: ohjelmamuistiin ja käyttömuistiin. Laitteiden ohjelmamuistiin ei tarvitse ladata erilaisia ohjelmia, joten ohjelmamuistin ei tarvitse olla käyttäjän eikä laitteen muutettavissa. Sulautetuille järjestelmille olennainen ominaisuus, on että mikrotietokoneen on säilytettävä muistiin tallennettu ohjelma, vaikka laitteen käyttöjännite katkeaisi. Tämän takia laitteiden tarvitsema ohjelma on tallennettu kiinteään haihtumattomaan ROM-muistiin. Tästä johtuu myös se että sulautetuissa järjestelmissä ei käytetä apumuistilaitteita laitteiden tarvitseman ohjelman tallentamista varten.

Mikrotietokone sisältää kolme perusosaa: mikroprosessorin, muistit ja liitäntäpiirit. Kyseisiä osia yhdistää kolme väylää: tietoväylä, osoiteväylä ja ohjausväylä. Väyläksi kutsutaan usean loogisesti samaan toimintaan kuuluvaa johtimen ryhmää. Näitä kolmea väylää kutsutaan mikrotietokoneen sisäisiksi signaaleiksi. Sisäisten signaalien avulla mikroprosessori ohjaa muiden mikrotietokoneen väyliin liitettyjen osien toimintaa. Toisaalta

mikroprosessorilla on ulkoisia ohjauksia, jotka ohjailevat prosessorin toimintaa. Ulkoiset ohjaukset vaikuttavat koko mikrotietokoneen toimintaan.

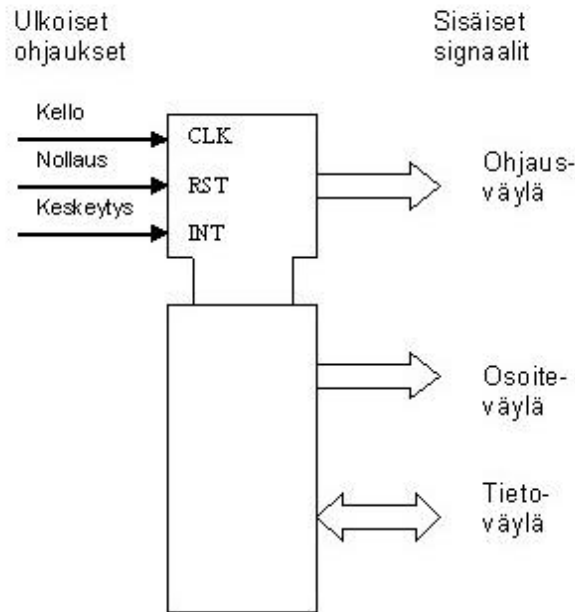
Mikroprosessorin toiminnan voi jakaa tarkastelun kannalta kahteen osaan. Mitä tapahtuu mikroprosessorin ulkopuolella ja toisaalta mitä tapahtuu sen sisällä. Mikroprosessorin ollessa mikrotietokoneen tärkein osa sen tehtävänä on lukea muistiin tallennettua ohjelmaa ja suorittaa siinä määritellyt toimenpiteet. Mikroprosessori lukee suurimman osan toiminta-ajastaan tietoa muistipiireistä. Lukutoiminto on suurimmaksi osaksi ohjelman lukua muistipiireistä, mutta mikroprosessori lukee myös liitäntäpiireistä tietoa. Mikroprosessori kirjoittaa välillä muisti- ja liitäntäpiireihin suorittaessaan ohjelmassa olevia toimintoja.

Mikrotietokoneen jokaisella väyliin liitettyllä muisti- ja liitäntäpiirillä on omat osoitteensa, joiden perusteella mikroprosessori voi lukea ja kirjoittaa niihin. Näistä osoitteista muodostuu osoiteväylä, joka on yksi suuntainen mikroprosessorilta lähtevä väylä. Prosessori määrää osoiteväylän avulla sen muisti- tai liitäntäpiirin sekä muistipaikan, jonka kanssa se asioi. Osoiteväylän leveydellä ei ole merkittävää vaikutusta prosessorin toimintaan. Osoiteväylän leveys määrää mikrotietokoneen muistiavaruuden suuruuden eli kuinka monta muistipiiriä mikroprosessoriin voidaan liittää. Jotta prosessori voi asioida muisti- ja liitäntäpiirien kanssa sekä lukea ohjelmakoodia ohjelmamuistista, tarvitaan kaksisuuntainen tietoväylä. Tietoväylän leveys määrittelee sen, kuinka paljon prosessori voi siirtää ja kuinka paljon käsitellä tietoa yhden muistiosoituksen avulla. Tämän takia tietoväylän leveydellä on hyvin suuri vaikutus prosessorin tehokkuuteen. Tietoväylä on yleensä 4, 8, 16, 32 tai 64 bittiä leveä. Vaikka tietoväylän leveys olisi pieni, voi sen avulla käsitellä isojaakin tietoja yhdellä osoituksella. Tällöin tietoa on kuitenkin paloitteltava ja se tuo haasteellisuutta ohjelmoiin. Mikrotietokone tarvitsee osoite- ja tietoväylän lisäksi väylän, joka kertoo muisti- ja liitäntäpiireille sen, miten prosessori asioi niiden kanssa. Tätä väylää kutsutaan ohjausväyläksi. Väylä on yksisuuntainen ja sen tärkeimmät signaalit ovat luku- ja kirjoitussignaalit. Näiden avulla prosessori määrittelee lukeeko vai kirjoittaanko se muisti- tai liitäntäpiireihin. Mikroprosessorin luku- ja kirjoituslähtöjen toteutuksessa on kaksi yleisimmin käytettyä tyyppiä. Näitä toteutustapoja kutsutaan Intel- ja Motorola-tyyppisiksi toteutuksiksi. Intel-tyyppisessä toteutuksessa sekä lukemista että kirjoittamista varten ovat

erilliset liitännät. Motorola-tyyppisessä toteutuksessa on yksi fyysinen liitäntä. Liitännällä kun on kaksi tilaa, Motorola-tyyppisessä toteutuksessa liitäntä on joko luku- tai kirjoittamistoiminnan kannalta aktiivisessa tilassa. Intel-tyyppisessä toteutuksessa liitännöistä voi olla joko luku- tai kirjoitusliitäntä aktiivisessa tilassa tai kumpikin voi olla ei-aktiivisessa tilassa.

Mikroprosessorin ulkoiset ohjaukset voivat olla hyvin erilaisia riippuen prosessorista. Yleisimmillä prosessoreilla ne ovat kuitenkin kuvan 10 mukaiset tuloliitännät. Synkroninen kellosignaali tahdistaa koko mikrotietokonetta. Prosessori voi sisältää itsessään sisäisen kello-oskillaattorin, jolloin käytetään ulkoista kidettä. Kiteen kanssa tarvitaan kaksi noin 10 - 32 pF kondensaattoria. Toinen mahdollisuus mikroprosessorin tahdistamiselle on kokonaan ulkoinen kello-oskillaattori. Tällöin käytetään nykyisin tietylle taajuudelle valmistettuja kideoskillaattorimoduuleita. Ulkoisia ohjauksia ovat myös asynkroniset nollaus- ja keskeytyssignaalit. Nollaussignaalin avulla saadaan prosessorin toiminta pysäytettyä ja aloitettua alusta. Kun prosessorin nollaustulo on aktiivinen, prosessorin toiminta pysäytetään ja sen sisäiset rekisterit asetetaan alkuarvoihin. Nollaustulon palautuessa lepotilaan prosessori aloittaa ohjelmansuorittamisen alusta tietyistä käynnistysosoitteesta.

Sulautetuissa järjestelmissä prosessori suorittaa päättymätöntä ohjelmasilmuksia. Yleensä mikrotietokoneen ohjaamassa prosessissa tulee tilanne joka vaatii välitöntä toimenpidettä. Esimerkiksi käyttäjä painaa laitteessa olevaa painiketta. Prosessorin ohjelmaa ei voida laatia yleensä siten että prosessori lukee säännöllisin väliajoin onko painiketta painettu. Tilanteet tapahtuvat yleensä hyvin harvoin ja ne ovat huonosti ennustettavissa. Tämän kaltaiset ongelmat voidaan ratkaista keskeytyksien avulla. Prosessorin saadessa keskeytystuloonsa aktiivisen tilan se siirtyy keskeytyspalveluohjelmaan (*Interrupt Service Routine*), jossa suoritetaan vaadittavat toimenpiteet. Tämän jälkeen prosessori palaa jatkamaan ohjelman suorittamista siitä mihin jäi saadessaan keskeytyksen. Keskeytyksiä on kahdenlaisia, on sekä ohjelmallisesti estettävissä olevia (*Interrupt, INT*) että ei ohjelmallisesti estettäviä (*Non-Maskable Interrupt, NMI*). Kuvassa 10 on esitetty myös mikroprosessorin sisäiset signaalit. [11, s.13 - 18]



Kuva 10. Mikroprosessorin ulkoiset ohjaukset ja sisäiset signaalit [11, s. 15]

Mikrotietokoneissa on haihtuvaa ja haihtumatonta muistia. Haihtuvaa eli RAM-muistia (*Random Access Memory*) mikroprosessori käyttää käyttömuistina, johon se tallentaa ohjelman suorittamisen aikana tarvitsemaansa tietoa. Tiedon ei tarvitse säilyä käyttöjännitteen katketessa. Haihtumatonta eli ROM-muistia (*Read Only Memory*) käytetään prosessorin ohjelman tallentamiseen.

Liitäntäpiirien avulla mikrotietokone on yhteydessä sitä ympäröivään muuhun elektroniikkaan. Liitäntäpiirit voidaan jakaa tulo- ja lähtöpiireihin. Näitä piirejä kutsutaan yleensä niiden vähyyden vuoksi rekistereiksi. Yksinkertaisimmillaan liitäntäpiiri sisältää kaksi rekisteriä. Tiedonsuunta- ja tietorekisterien avulla määritellään kumpaan suuntaan ja mitä tietoa liikutetaan.

Mikrotietokoneet voivat sisältää monenlaisia haihtumattomia muisteja, joista yleisimpiä ovat ultraviolettivalolla purettava lukumuisti EPROM ja OTP-EPROM (*One Time Programmable, OTP*). EPROM muistityyppi on uudelleen ohjelmoitavissa. Tämän muistityypin omaavien mikrotietokoneiden kotelossa on ikkuna, jonka kautta voidaan ultraviolettivalolla tyhjentää sen muisti. Koska EPROM muistityypin uudelleen ohjelmointi ominaisuutta käytetään harvoin, on kehitetty huomattavasti halvempi ikkunaton kotelotyyppin mikrotietokone. Tämän tyyppin mikrotietokone ei ole uudelleen

ohjelmoitavissa. Sähköisesti purettavat EEPROM ja Flash-tyypin muistit ovat yleistyneet kovasti viime vuosina. Nämä muistityypit ovat vielä hyvin kalliita, joten eivät ole niin yleisiä kuin OTP-EPROM ja EPROM-muistit. Haihtuvista muisteista yleisin sulautetuissa järjestelmissä on staattinen luku-kirjoitusmuisti (*Static Random Access Memory, SRAM*).

4.2 Yleistä ohjelmoinnista

Mikrotietokoneita eli sulautettuja järjestelmiä ohjelmoitaessa voidaan käyttää eri ohjelmointikieliä. Suurin osa nykypäivän sulautettujen järjestelmien ohjelmoinnista tapahtuu laiteläheisellä C-kielellä. Myös muita korkean tason kieliä käytetään, mutta vähissä määrin. C-kielen syntaksi on kehitynympäristöriippuvainen kun assembler-kieli on prosessoririippuvainen.

Vaikka ohjelmointi tapahtuukin nykyään pääosin C-kielellä, on ohjelmoijan tunnettava kuitenkin assembler-kielen perusteet. Prosessorien rekistereitä ei voi ohjata puhtailla C-kielisillä käskyillä vaan se tapahtuu assembler-kielisillä käskyillä joita on kirjoitettu C-kielisen ohjelman sisään kehitynympäristön vaatimalla syntaksilla. Myös laitesuunnittelijan on tunnettava assembler-perusteet, jotta yhteistyö ohjelmoijan kanssa sujuisin paremmin.

Assembler-kieltä kutsutaan symboliseksi konekieleksi. Jokaisella prosessorilla on oma konekielensä eli jollekin prosessorille tehty konekielinen ohjelma ei toimi täysin toisella prosessorilla. Monet prosessorivalmistajat tekevät nykyään prosessoreistaan samaa konekieltä tukevia kuin niiden aikaisemmin valmistamat prosessorit. Konekielinen käsky määrittää prosessorille hyvin yksiselitteisesti mitä sen pitää tehdä. Käskyt sisältävä operaattorin, joka kertoo mitä tehdään ja operandin, joka kertoo mille rekisterille tai muistipaikalle toiminto kohdistuu. Käskyt voivat sisältää yhden tai useamman operandin. On myös olemassa käskyjä jotka eivät sisällä ollenkaan operandeja. Näitä ovat esimerkiksi käsky, joka pysäyttää prosessorin, käsky, joka kääntää prosessorin menemään lepotilaan sekä käsky, joka palauttaa ohjelman suorittamisen keskeytyspalvelusta takaisin ohjelman suorittamiseen. Vanhemmissa prosessoreissa on yleensä yksi rekisteri, joka on tärkeämpi kuin yleiskäyttöiset rekisterit. Tätä rekisteriä kutsutaan yleensä A-rekisteriksi ja siitä käytetään myös nimitystä akku. A-rekisteri on yleisesti se rekisteri, joka on oltava toisena operandina lasku- ja

päättelyoperaatioissa. Mikrotietokoneen muut rekisterit ovat yleiskäyttöisiä 8-bittisiä rekistereitä.

Prossessorin käskykannaksi kutsutaan niiden operaatiokoodien joukkoa, mitkä ovat käytössä kyseisellä prosessorilla. Jokaisella prosessorilla on omanlainen käskykanta. Käskykantojen suuruus vaihtelee prosessoreittain muutamasta kymmenestä muutamaa sataan operaatiokoodiin. Konekieliset käskyt ovat hyvin vaikea muistaa, joten on kehitetty symbolinen konekieli. Symbolisen konekielen kehittämisen on vaikuttanut myös konekielisen ohjelmoinnin työläys. Symbolisessa konekielessä on annettu konekielisille käskyille lyhyempi nimi; 2 - 4 merkkiä pitkä. Myös symbolinen konekieli on erilainen eri prosessoreille. Symbolisen konekielen käskyjen nimet tulevat suoraan niiden toiminnallisesta englanninkielisestä nimestä.

Käskyt voidaan jakaa yleisellä tasolla paikasta toiseen siirtäviin tiedonsiirtoikäskyihin, aritmeettis-loogisiin tiedonkäsittelykäskyihin ja ohjaus- ja haarautumiskäskyihin. Tiedonkäsittelykäskyt voidaan jakaa vielä neljään alaryhmään: aritmeettiset käskyt, loogiset käskyt, siirto- ja pyörittämiskäskyt ja vertailukäskyt. Aritmeettisillä käskyillä toteutetaan kaikki laskenta. Loogisilla käskyillä operoidaan biteittäin rekistereissä olevaa tietoa kun taas siirto- ja pyörittämiskäskyillä siirretään bittejä rekistereissä yhdellä joko oikealle tai vasemmalle. Vertailukäskyillä puolestaan verrataan yleensä akun sisältöä johonkin rekisterin, muistipaikan tai vakion arvoon. Prossessorin ohjelma voi haarautua joko ehdottomasti tai ehdollisesti. Haarautuminen voi tapahtua absoluuttisesti tai suhteellisesti. Absoluuttisesti haarautuessa operaattorille annetaan suoraan osoite, josta prosessori jatkaa ohjelman suorittamista. Suhteellisesti haarautuessa annetaan operaattorille hypyn pituus. Suhteellisessa hypyssä operandi voi olla joko positiivinen tai negatiivinen. Prossessorin lippurekisterien lippujen asetus- ja nollauskäskyjä, keskeytysten esto- ja sallintakäskyjä, pysäytys- ja nukutuskäskyjä sekä tyhjäksiä kutsutaan prosessorin ohjauskäskyiksi. [11, s.93 - 99]

Prossessorille kirjoitettu ohjelma käännetään aina konekieliseksi riippumatta siitä, millä kielellä se on kirjoitettu. Suurempia ja monimutkaisempia ohjelmia kirjoitetaan sovelluskehittämiä apuna käyttäen. Yksinkertaista ohjelmaa tehtäessä symbolisella konekielellä kirjoitettuna ohjelmoija pystyy tekemään ohjelmasta lyhyemmän ja nopeampaa koodia kuin lausekielestä kääntäjän tekemä konekieli. Tämä on syynä siihen että vielä käytetään symbolista

konekieltä joko osana lausekieltä tai kokonaan ohjelmoitaessa mikrotietokoneita.

Ohjelmointityön jälkeen kirjoitetun koodin kääntäminen ja simulointi, jos sellaiseen on kehitynympäristössä mahdollisuus, on erittäin suotavaa ja myös kustannuksiltaan edullisempaa. Näin säästyy prosessoreita turhilta ohjelmoinneilta, varsinkin jos prosessorin muisti on OTP (*One Time programmable*) -tyyppinen.

Tässä projektissa päädyttiin ohjelmoimaan assembler-kielellä. Ohjelmointiympäristönä oli STMicroelectronicsin ST6-prosessoriperheelle tehty RIDE-ohjelmisto.

4.3 Vuokaavio

Ohjelman suunnittelu alkaa assembler-kielellä kirjoitettaessa vuokaavion tekemisellä. Tämän työn yhteydessä laaditun vuokaavion lopullinen versio on esitetty liitteessä 3. Vuokaaviossa käytetään pääsääntöisesti kolmea erilaista lohkoa. Toiminnan alku merkitään päistä pyöristetyllä suorakulmiolla. Lisäksi vuokaaviossa käytetään toimintalohkoa, joka on suorakulmio ja päättelylohkoa, joka kärjellä oleva suunnikas.

Luvussa 2 olevan spesifikaation mukaisesti ohjelmalla on kaksi päähaaraa. Laitteella on pystyttävä tekemään kalibrointi ja testaus annettaessa ulkoinen ohjaussignaali siitä kumpaa halutaan tehdä. Spesifikaatiossa määritelty kuukausittain tehtävän automaattisen testin ajoittamiseen kuluu suurin osa prosessorin ohjelman suorittamiseen käyttämästä ajasta. Kalibrointi- ja testauskäskyn antaminen prosessorille onnistuu laskentatoiminnon aikana vain jos mikrotietokoneen liitäntäpiirit alustetaan inputeiksi. Laskennan aikana pitää pystyä antamaan joko kalibrointi tai testauskäsky. Laskennan loppuun suorittamisen jälkeen siirrytään automaattisesti suorittamaan testiä.

Kalibrointihaaran suorittamien on hyvin suoraviivainen. Ensimmäisenä on alustettava mikrotietokoneen liitäntäpiirit kalibrointi varten. Kalibroinnin suorittamisen edellytyksenä on kaikkien turva- ja opastevalojen oltava päällä. Valojen sytyttämisestä kauko-ohjauksen välityksellä annetun käskyn jälkeen on odotettava valojen syttymistä. Tämän jälkeen täytyy valita kanava jolle tehdään kalibrointi. Prosessori pystyy käsittelemään vain yhtä kanavaa kerrallaan. Kanavan valinnan jälkeen suoritetaan kyseisen kanavan

analogisella jännitteelle A/D-muunnos. Tämä tulos on tallennettava muistiin myöhempää testaustoimintaa varten. Vuokaaviossa tullaan päättelylohkoon, jossa katsotaan onko kaikille kahdeksalle kanavalle suoritettu kalibrointi. Jos kanavia on käymättä läpi, mennään EI-nuolen suuntaan. Muussa tapauksessa siirrytään seuraavaan toimintalohkoon eli annetaan käsky kauko-ohjauksen välityksellä valojen sammuttamista. Kalibroinnin suorittamisen jälkeen siirrytään alustamaan liitäntäpiirit jälleen laskennan aikaiseen tilaan.

Testaushaara on monimutkaisempi kalibrointiin verrattuna. Toiminta haarojen viisi ensimmäistä toimintalohkoa ovat samanlaiset. Nyt saatua A/D-muunnoksen tulosta ei tallennetakaan muistiin vaan verrataan muistissa olevaan kalibroinnin yhteydessä tallennettuun tietoon. Vikatilanne päättelylohkossa katsotaan onko muistissa oleva arvo isompi kuin siihen verrattava arvo. Jos muistissa on isompi arvo kuin tämän hetkinen kanavasta saatu arvo, annetaan vikahälytys. Muuten siirrytään päättelylohkoon, jossa katsotaan onko kaikki kanavat käyty läpi. Kaikkien kanavien testauksen jälkeen siirrytään seuraavaan päättelylohkoon, jonka perusteella mennään joko testin alkuun tai annetaan kauko-ohjauksen kautta käsky sammuttaa valot. Testin suorittaminen kokonaisuudessaan pitää spesifikaation mukaan kestää useita minuutteja. Myös huoltomiehen on ehdittävä näkemään EXICMD-kortin led-valot, jos testauksessa on huomattu vika. Testin suorittamisen jälkeen palataan ohjelman suorittamisessa alkuun tai aikalaskentaan sen mukaisesti onko testitilaan tultu ulkoisen ohjauksen vai loppuun suoritettun laskennan seurauksena.

4.4 Koodin kirjoittaminen

Hyvän vuokaavion laatiminen helpottaa assembler-kielellä kirjoitettaessa koodin kirjoittamista. Koodin kirjoittamisessa oli tukeuduttava prosessorin valmistajan julkaisemaan koodimaniaaliin, jossa oli prosessorin ymmärtämien käskyjen kaikki käskymuodot. Tämän manuaalin ja prosessorin datalehden avulla ohjelmointi onnistui hyvin.

Projektissa kirjoitettu koodi ei ole julkinen, joten sen tarkka läpikäynti tässä kirjallisessa työssä ei ole mahdollista. Koodi kirjoitettiin järjestykseltään hiukan vuokaavion esittämästä järjestyksestä poiketen. Ensimmäisenä koodissa on porttien ja rekisterien alustus kalibrointi varten.

Kalibroitisegmentin koodi on kuitenkin vasta testaussegmentin jälkeen. Aikalaskennan koodi on kalibroitisegmentin jälkeen ennen porttien ja rekisterin alustusta, joka suoritetaan heti prosessorin käynnistymisen jälkeen. Lopuksi ennen keskeytysvektorien määrittelyä on testausta varten tarvittavat porttien ja rekisterien alustukset.

Koodia kirjoittaessa oli mahdollisimman usein nollattava vahtiajastin, jotta vahtiajastin ei nollaa missään vaiheessa prosessoria. Vahtiajastin saatiin nollattua lataamalla WDR rekisteriin heksadesimaalinen luku 0FEh. Eli tässä laitteistossa vahtiajastimella ei ollut mitään käyttöä.

Kalibroitisegmentissä kuten testaussegmentissä jokaiselle kanavalle tehtävä toimenpide kirjoitettiin erikseen. Segmentin ensimmäinen toimenpide oli antaa käsky kauko-ohjaukselle sytyttää valot. Kanavien valinta tapahtui prosessorin B-portin OR (*Option Register*) -rekisterin avulla asettamalla rekisterin kanavaa vastaava bitti ykköseksi. Seuraavaan kanavaan siirryttäessä oli nollattava edellisen kanavan valinta. Kanavavalinnan jälkeen annettiin A/D-muuntimelle käsky suorittaa muunnos. Koska mitattavien lamppujen virta voi laskea niiden lämmetessä ja vanhetessa, A/D-muuntimelle tuleva jännite voi samasta syystä myös laskea. Tämä täytyi ottaa huomioon tallennettaessa tulosta muistiin. A/D-muuntimen ollessa lineaarinen 8-bittinen muunnin 0 voltista +5 volttiin, muunnoksen tulosta voitiin dekrementoida, eli vähentää yksi, viidesti. Näin saatiin tallennettua sellainen arvo että lamppujen kuluttama virta voi vähentyä melkein yhtä paljon kuin yksi lamppu kuluttaa, mutta laitteisto ei anna siitä vielä vikailmoitusta. Mutta jos yksi lamppu menee rikki, virta laskee enemmän kuin mitä viidesti dekrementointi aiheuttaa muunnoksen arvoon. Tulos tallennettiin prosessorin muistiin alueelle, joka on varattu käyttömuistiksi ja on haihtuvaa muistia. Muistiosoitukset annettiin absoluuttisina, jotta niiden hakeminen muistista testaussegmenttiä kirjoittaessa on helpompaa ja yksinkertaisempaa.

Testaussegmenttiä kirjoitettaessa valojen sytyttämisen määräävän käskyn jälkeen valittiin ensiksi testattava kanava ja annettiin A/D-muuntimelle käsky suorittaa muunnos. Koska prosessorilla on A-rekisteri ja sen on oltava toisena operandina loogisissa operaatioissa, ladattiin muistiin tallennettu kanavaa vastaava lukuarvo siihen. A/D-muunnin tallentaa muunnostuloksen omaan ADR-rekisteriinsä. Akussa olevaa arvoa verrattiin

muunnostulokseen. Jos muunnos oli pienempi kuin akun arvo, ladattiin A-portin DR (*Data Register*) -rekisteriin arvo, josta led-valojen ohjauspiiri ymmärsi mikä led pitää sytyttää. Jotta led kerkeää syttymään täytyi vikailmoituksen perään saada viivettä riittävästi. Jos virhettä ei kuitenkaan ilmennyt, annettiin absoluuttinen hyppykäsky mennä suoraan seuraavaan kanavaan. Tämä edellä kuvattu testimenetelmä kirjoitettiin jokaiselle kanavalle erikseen. Lopuksi testisegmentistä muodostettiin silmukka jonka suorittaminen prosessorilta kestää muutamia minutteja.

Ohjelman koodista kaikki keskeytysvektoreita lukuun ottamatta määriteltiin RSEG-segmentin sisään. Keskeytysvektoreille oli CSEG-niminen segmenttimäärittäjä. Jokaiselle mahdolliselle keskeytysvektorille oli määriteltävä oma toiminta. Keskeytyksille, joita ei otettu käyttöön, kirjoitettiin vain käsky palata takaisin ohjelman suorittamiseen. Tässä laitteistossa tarvittiin käyttöön kolme keskeytysvektoria. Kaikissa sulautetuissa järjestelmissä ainakin RESET-keskeytykselle pitää määritellä toiminta eli määrittää prosessorin aloittamaan ohjelman suorittaminen alusta. Nyt otettiin myös NMI-pinnille ja A-portin ykköspinnille tuleva keskeytys käyttöön. NMI-keskeytyksen saatuaan prosessori määrättiin siirtymään kalibrointisegmenttiin. Mikrotietokoneen NMI -pinni kytkettiin kalibrointi-nappiin. A-portin ykköspinnin kautta testausnapilta tullut keskeytys määriteltiin määräämään prosessori suorittamaan testausta. Keskeytysvektorien määrittämisessä pitää määritellä myös missä muistiosoitteessa kyseinen keskeytysvektori sijaitsee.

Kirjoitettu koodi testattiin modifioidulla prototyyppi-piirilevyllä. Prototyyppi-piirilevyyn tehtiin ensin tarvittavan elektroniikkamuutokset ja valmis koodi voitiin testata näin saadussa lopullisen piirilevyn kaltaisessa laitteistossa.

syttyä EXICMD-piirikortilla. Tällä todennetaan samalla se, että kalibrointi on onnistunut. Kesken testin voi ottaa yhden lampun pois mittauskytkennästä, joka on sama asia kuin yksi lamppu olisi palanut. Tällöin pitäisi jokaisen ledin EXICMD-levyllä alkaa vilkkua vuorotellen. Tällä todennetaan samalla se, että laite kykenee erottamaan yhden lampun. Testi aika on noin 6 minuuttia, jonka jälkeen laite palaa odotustilaan.

Lopuksi suoritetaan niin sanottu nollakalibrointi jolla testataan se, että laite ei anna vikailmoitusta, jos jokin lamppukanavista on käyttämättä. Kun edellinen testi on suoritettu loppuun, tehdään uudelleen kalibrointi niin että lamppujen syöttö on katkaistu kokonaan. Virtamuuntajille ei mene ollenkaan virtaa. Kalibroinnin jälkeen tehdään testaus. Yhdenkään ledin ei kuulu syttyä tässä testissä.

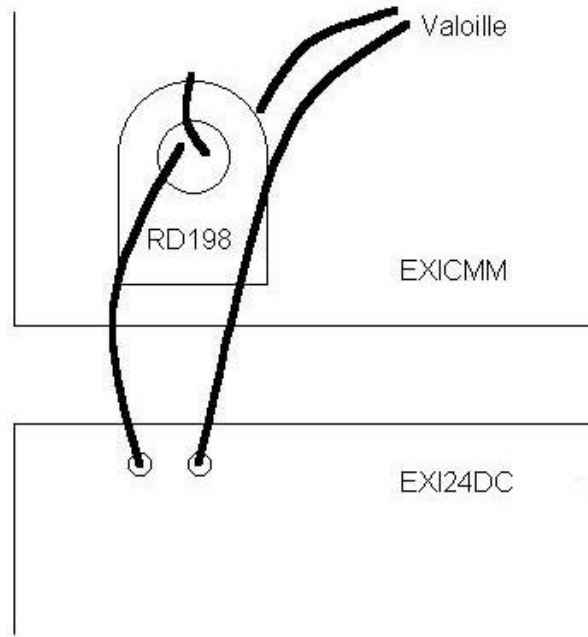
6 LAITTEISTON ASENNUS JA KÄYTTÖ

6.1 Asennus

Ryhmävalvontakeskuksen piirikortit yhdistetään toisiinsa 16-karvaisella lattakaapelilla, jossa on kummassakin päässä Molexin 16-pinniset liittimet. Tämän jälkeen turvalokeskuksesta lähtevien syöttökaapelien toinen karva kierretään sitä vastaavan EXICMM-kortilla olevien virtamuuntajien läpi kuvan 12 osoittamalla tavalla.

Tämän jälkeen kytketään EXICMM-kortin vasemmassa yläreunassa oleva kauko-ohjausliitin johtimella turvalokeskuksen kytkinliittimeen. Katso kuva 9. Johtimet kytketään joko rinnan- tai sarjankytkentäliittimeen riippuen siitä toimiiko kauko-ohjaus aukeavalla vai sulkeutuvalla kosketintiedolla. Jos kauko-ohjaus toimii avautuvalla kosketintiedolla, kytketään se sarjankytkentäliittimeen ja jos se toimii sulkeutuvalla kosketintiedolla, kytketään se rinnankytkentäliittimeen.

Vielä ennen jännitteiden kytkentää kytketään EXICMM-kortin vasemmassa reunassa oleva ALARM-liitin johtimella sinne mihin halutaan kosketintieto saada. Tämän jälkeen laitteistolle voidaan kytkeä 24 voltin DC-jännite EXICMM-kortin oikeassa reunassa olevaan liittimeen. Laitteiston käyttöjännite ei saa katketa missään vaiheessa. Jännitteen katkeaminen aiheuttaa prosessorin nollauksen ja siinä tapauksessa automaattinen kuukausitestaustilasto menee sekaisin ja alkaa alusta sekä tallennetut kalibrointitiedot häviävät.



Kuva 12. Lamppujen syöttöjohtojen asentaminen virtamuuntajiin.

6.2 Käyttö

Asennuksen jälkeen ensimmäinen toimenpide on tehdä kalibrointi painamalla kalibrointi-nappia EXICMD-kortilta. Kalibrointi kestää noin 45 sekuntia.

Laitteistolla voidaan tehdä testaus valoille ihan milloin vain painamalla EXICMD-kortilta testinappia. Testi kestää noin 6 minuuttia. Laitteisto laskee jatkuvasti kuukauden kestäväää laskentaa, mutta se ei vaikuta testin tekemiseen. Laskennan jälkeen laitteisto tekee automaattisesti testin.

Sekä kalibroitaessa että testattaessa laitteisto antaa EXI24DC-kortille kauko-ohjausliittimen välityksellä käskyn sytyttää valot.

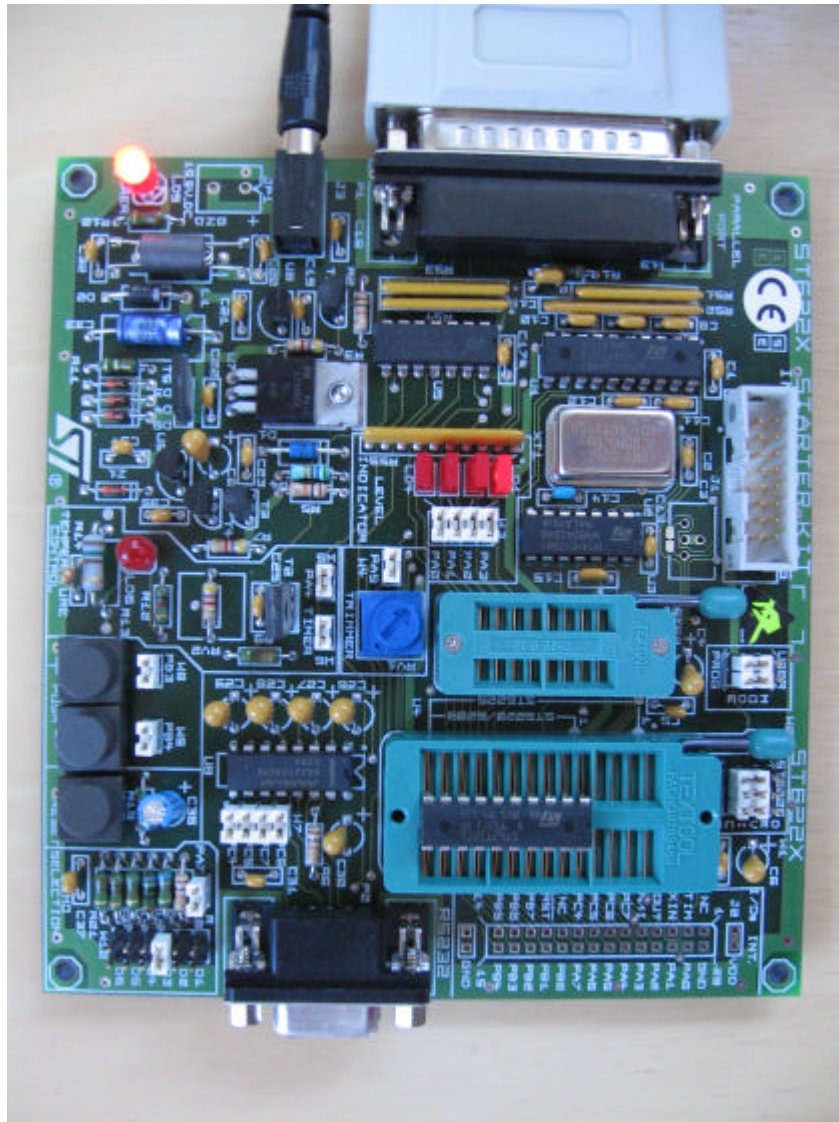
Jos lamppujen määrää muutetaan kanavilla, kalibrointi täytyy tehdä uudelleen. Sen voi tehdä välittämättä kuukausilaskennasta painamalla jälleen EXICMD-kortilta kalibroitinappia silloin kun testitoiminto ei ole käynnissä.

7 SUUNNITTELUYMPÄRISTÖT

Elektroniikkasuunnittelussa käytettiin yleisiä suunnittelutyökaluja. Kytkentöjen simuloinnissa käytettiin OrCADin Capture CIS -ohjelmistoa. Piirilevysuunnittelussa käytettiin PADS-ohjelmistoja, piirikaaviosuunnittelussa PowerLogicia ja piirilevyn layoutin tekemiseen PowerPCB:tä. Tässä työssä ei esitellä tämän tarkemmin näitä yleisesti käytössä olevia piirilevysuunnitteluohjelmistoja.

Proessorin ohjelmiston kehittämiseen käytettiin Raisonancen valmistamaa RIDE-kehitysympäristöä. Tämä ohjelmointikehitysympäristö kuului STMicroelectronicsin toimittaman kehitysalustan ST622x Starter Kit mukaan. Ohjelmisto on esitetty luvussa 7.1. Luodun ohjelman syöttämiseen prosessorille tarvittiin oma ohjelma, Winee -niminen Windows Epromer. Kuvassa 13 on kuvattuna STMicroelectronicsin valmistama kehitys- ja ohjelmointialusta. Alustan avulla pystyi ohjelmoimaan ST6-prosessoriperheeseen kuuluvia ST620x-, ST621x- ja ST622x-prosessoreja. Starter Kit -paketin mukana tuli myös yksi valmiiksi ohjelmoitu demoprosessori, jota pystyi ajamaan alustalla. Prosessorilla oli kolme demo-ohjelmaa. Kehitysalusta liitettiin tietokoneeseen LPT1-portin kautta.

Starter Kit -paketin sisältämistä kehittämistä ja alustasta on saatavilla myös kaikille STMicroelectronicsin prosessoriperheille sopivat versiot.



Kuva 13. STMicroelectronicsin ST622x Starter Kit kehitysalusta

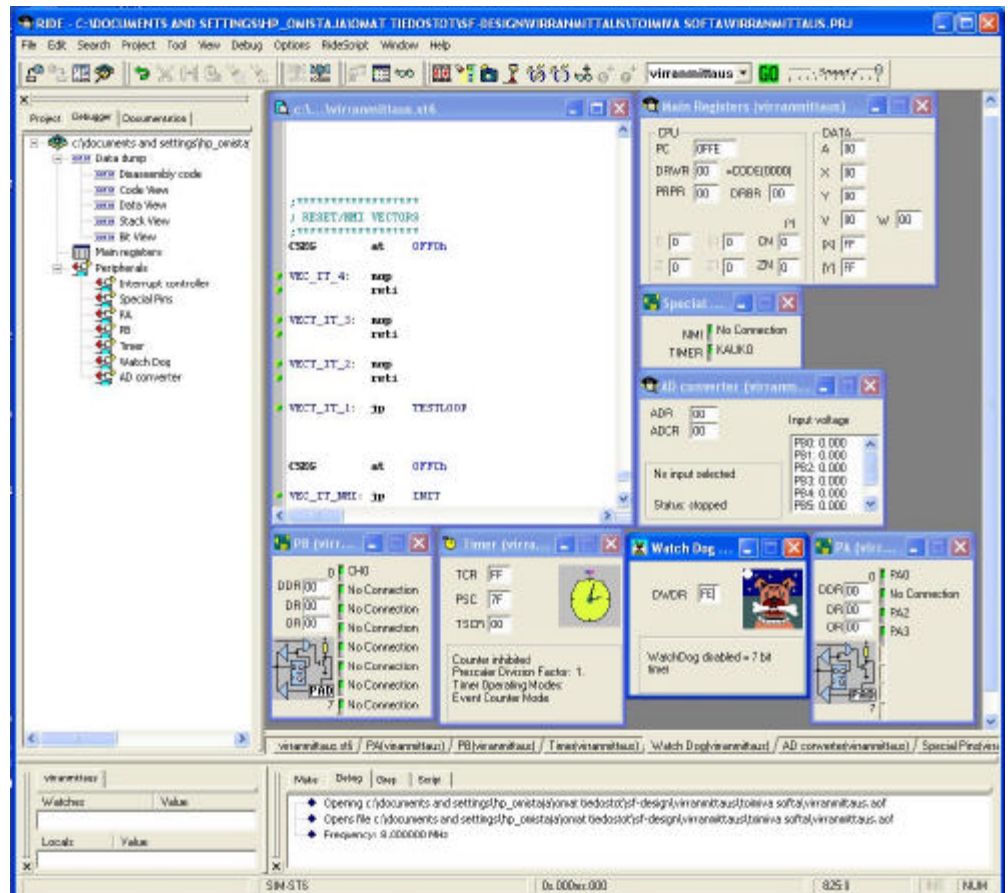
7.1 RIDE

Ohjelmistokehitysympäristön on kehittänyt Raisonance S.A.S.. Ohjelmisto on saatavilla valmistajan kotisivuilta lisenssittömänä versiona. Lisenssittömässä versiossa on rajattu laadittavan ohjelman kooksi 512 tavua. Lisenssin hankkimalla ohjelmistosta saa rajattoman version käyttöön.

Kehitysympäristö sisältää assembler- ja c-kielen kääntäjät. Lisäksi ohjelmistolla pystyi generoimaan hex-tiedostoja. Hex-tiedoston generointia tarvitaan, jotta laadittu ohjelma voidaan siirtää prosessorille.

Ohjelmistossa on tekstipohjainen ympäristö, jossa koodi kirjoitetaan. Kirjoitetun ohjelman tallennusmuoto riippuu käytettävästä kielestä. Tässä projektissa koodi tallennettiin assembler-tiedostona nimellä nimi.st6.

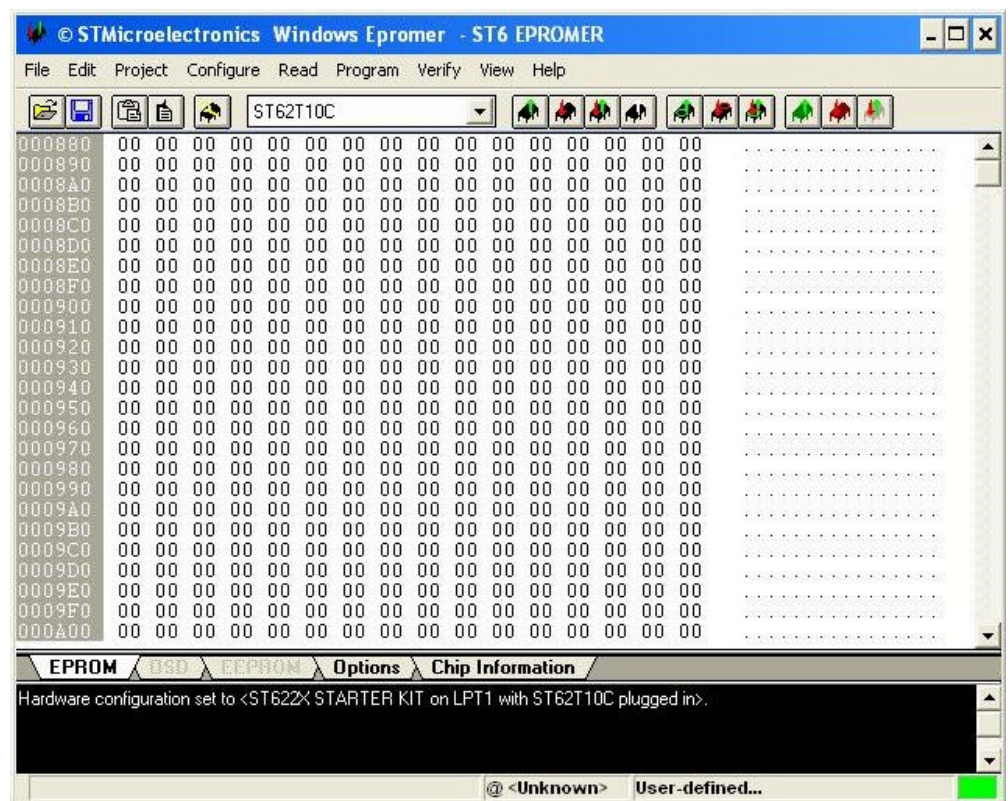
Kuvassa 14 on kehitysympäristön simulointinäkymä. Simulointiympäristössä pystyi säätämään simulointinopeuden ja seuraamaan jokaisen käskyn suorittamista debug-ominaisuuden ansiosta. Kaikkien rekisterien tilat saatiin näkyviin ja niiden muutoksia pystyttiin seuraamaan. Simuloitaessa koodia pystyttiin antamaan myös koodille ulkoisia ohjauksia. Prosessorin A/D-muuntimelle tulevia jännitteitä oli myös mahdollista asettaa. Kehitysympäristön käyttömanuaali oli ohjelmiston sisällä help-tiedostona.



Kuva 14. RIDE-ohjelmiston simulointinäkymä

7.2 Winee

RIDE-ohjelmistolla generoidun hex-tiedon lataamiseen prosessorille tarvittiin erillinen ohjelma. Tämän Winee-nimisen Windows Epromer -ohjelman on tehnyt STMicroelectronics. Ohjelmistolla pystyi lataamaan tietoa kumpaakin suuntaan sekä koko muistiavaruuden tai vain osan siitä. Ohjelmistolla pystyi myös verifioimaan muistiavaruuden tai vain osan siitä. Tällä ohjelmistolla määriteltiin prosessorin Options Register -rekisterin tiedot. Toisin sanoen ovatko keskeytykset ylipäänsä mahdollisia, käytetäänkö ohjelmallista vai ulkoista vahtiajastinta ja lukusuojausta sekä käytetäänkö ulkoista vai sisäistä kello-oskillaattoria. Kuvassa 15 on esitetty ohjelmiston työskentelynäkymä.



Kuva 15. STMicroelectronicsin valmistama laaditun ohjelman latausohjelma

Ohjelmoitaessa prosessoria Winee-ohjelmistolla tarkastettiin ensin prosessorin muistiavaruus, että se on tyhjä. Tämän jälkeen käskettiin siirtää hex-tiedosto prosessorille ja lopuksi suoritettiin verifiointi muistiavaruudelle.

8 YHTEENVETO

Asiakkaan toimittaman lopullisen tuotteen toiminnan testauksessa todettiin suunnittelutyön onnistuneen hyvin. Elektroniikan osalta laitteistosta saatiin suunniteltua valmistuskustannuksiltaan hyvin edullinen. Asiakkaan toimittama spesifikaatio pystyttiin täyttämään muutamaa muutosta lukuun ottamatta. Muutokset tehtiin asiakkaan hyväksynnän jälkeen. Myös asiakas täydensi suunnitteluprosessin aikana vaatimuksiaan. Laitteistoon piti saada indikointi-led-valot testaus ja kalibrointi toiminnoille. Laitteistoon lisättiin myös toinen painokytin, jotta laitteiston toimintamuodot saatiin toimimaan halutulla tavalla.

Ohjelmistokehityksessä vuokaavion käyttö oli edellytys ohjelmoinnin onnistumiselle. Ohjelmistokehitys oli projektissa työläin vaihe. Ohjelmointi saatiin kuitenkin onnistuneesti suoritettua ja ohjelmasta saatiin halutunlainen. Koodin kirjoittamisessa apuna oli suuresti kehitysympäristössä ollut simulointiympäristö. Simuloinnin avulla saatiin myös asiakkaan edullisuusvaatimus täytettyä. Simuloinnin avulla säästettiin suuri määrä prosessoreita.

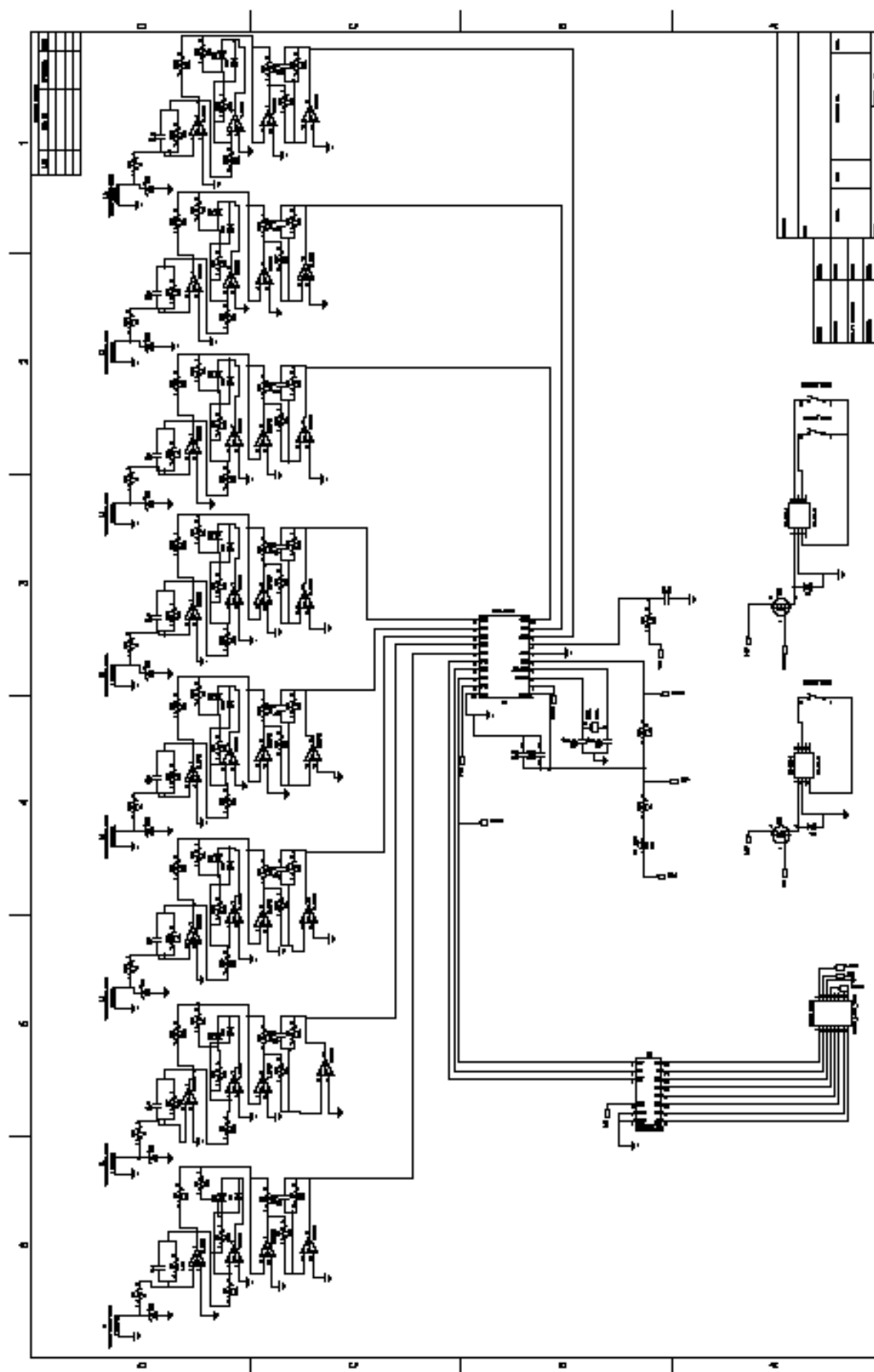
Projekti oli oppimisen kannalta hyvin opettavainen ja sopivan yksinkertainen ensimmäiseksi kokonaiseksi suunnitteluprojektiksi. Projektin myötä opittiin elektroniikkasuunnittelun perusteita ja ohjelmistokehityksen perusteet käytännössä sekä ohjelmoimaan assembler-kielillä. Assembler-kieli on yksinkertainen ja looginen kieli ohjelmoida yksinkertaisia ja pieniä ohjelmia mikrotietokoneelle. Symbolisella konekielellä ohjelmoitaessa kirjoitetusta koodista näkee ja pystyy ymmärtämään mitä prosessori missäkin vaiheessa tekee. Suuremmissa ohjelmissa on kuitenkin viisaampaa käyttää graafisia kehitysympäristöjä tai korkean tason kieltä. Korkean tason kielestä ei kovin helposti näe, mitä prosessori tekee käskyn suorittamisen aikana.

VIITELUETTELO

- [1] Euroopan unionin virallinen lehti, Euroopan parlamentin ja neuvoston direktiivi 2002/95/EY, 13.2.2003, s.19 - 23
- [2] ST6210C Prosessorin datalehti, [PDF -dokumentti]. lokakuu 2003, [viitattu 8.8.2006] Saatavissa: <http://www.elfa.se/pdf/73/736/07363708.pdf>
- [3] TL074CN -operaatiovahvistinpiirin datalehti, [PDF -dokumentti]. marraskuu 2001, [viitattu 8.8.2006]
Saatavissa: <http://www.elfa.se/pdf/73/731/07311707.pdf>
- [4] HC49/4H -kiteen datalehti, [PDF -dokumentti]. [viitattu 11.8.2006]
Saatavissa: <http://www.elfa.se/pdf/74/07451503.pdf>
- [5] 74HC138N -piirin datalehti, [PDF -dokumentti]. 23.12.2005, [viitattu 8.8.2006] Saatavissa:
http://www.semiconductors.philips.com/acrobat/datasheets/74HC_HCT138_3.pdf
- [6] GS-SH-203T -releen datalehti, [PDF -dokumentti]. [viitattu 11.8.2006]
Saatavissa: <http://www.goodsky.co.uk/datafiles/GS.pdf>
- [7] BSN304 -transistorin datalehti, [PDF -dokumentti]. helmikuu 1995 [viitattu 19.8.2006] Saatavissa: <http://www.elfa.se/pdf/71/07107782.pdf>
- [8] LMC7660IN -jänniteinvertioijan datalehti, [PDF -dokumentti]. helmikuu 2005, [viitattu 19.8.2006] Saatavissa: <http://www.elfa.se/pdf/73/731/07314107.pdf>
- [9] Hambley, Allan R., *ELECTRONICS; A Top-Down Approach to Computer-Aided Circuit Design*. Printed in the USA: Pentice-Hall, Inc. 1994
- [10] Vehmas, Timo, *Elektroniikka-askartelu: Pohjaksi atomiajan harrasteeseen*. Porvoo, Helsinki: WSOY. 1971
- [11] Koskinen, Jari, *Mikrotietokonetekniikka, Sulautetut järjestelmät*. 1. uudistettu painos. Keuruu, Otava. 2004 (1998)

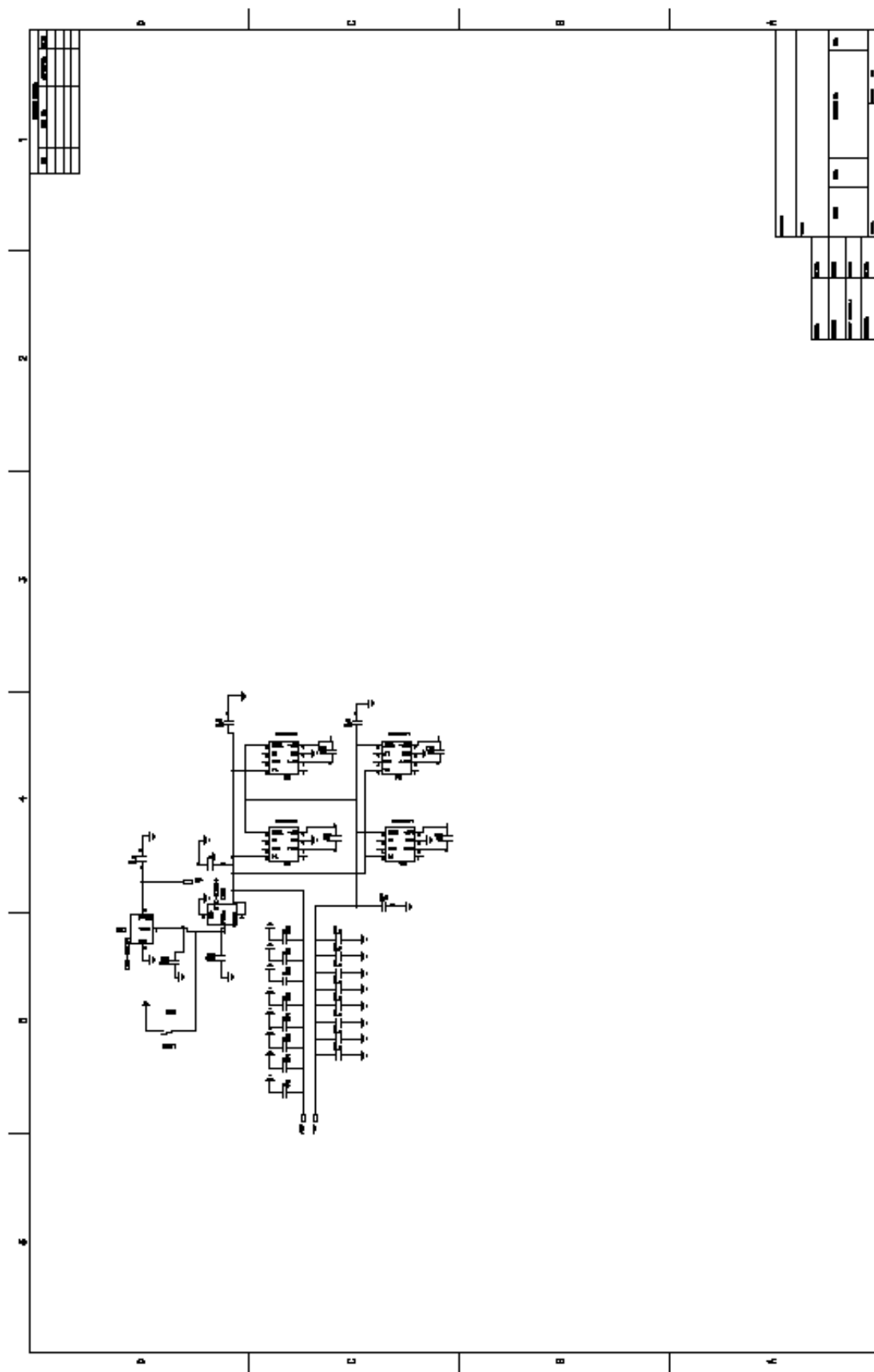
EXICMM -piirikortin piirikaaviot

EXICMM103.sch-1 - Wed Aug 02 09:23:34 2006



EXICMM-piirikortin toiminnallisenosan piirikaavio

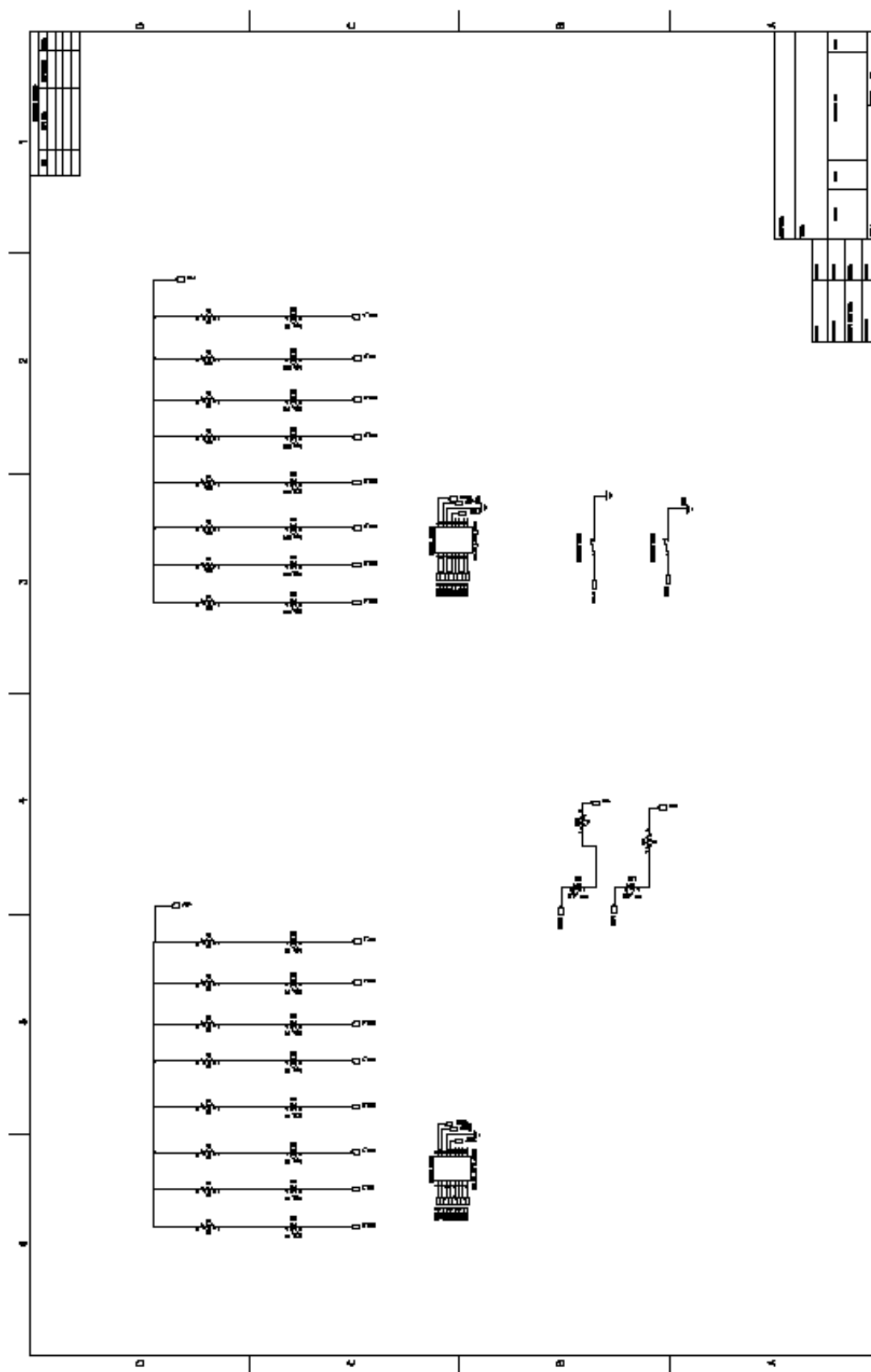
EXICMM103.sch-2 - Wed Aug 02 09:23:53 2006



EXICMM-piirikortin jännitesyöttöosan piirikaavio

EXICMD-piirikortin piirikaavio

EXICMD102.sch-1 - Wed Aug 02 09:24:46 2006



Assembler-koodin vuokaavio

